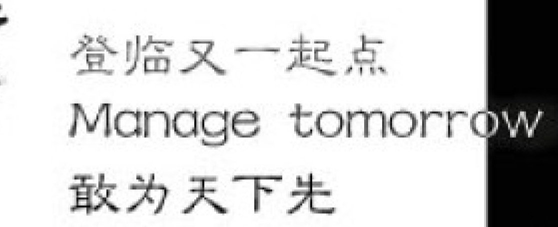
在云端

云计算最佳实践



胡嘉玺 编著



智慧 VMware vSphere 运维实录

清华大学出版社

虚拟智慧

VMware vSphere 运维实录

胡嘉玺　**编著**

清华大学出版社

北　京

内 容 简 介

VMware 跨时代产品 vSphere 4.x 是全世界第一个云端操作系统，并且真正将硬件 CPU、内存、存储、网络设备等设备抽象化，成为一个可计量单位，从而将这个计量单位转换成服务性质，让 IT 人员更方便地设计自身企业的平台架构，或是作成新的产品服务。

本书是全球第一本将所有 vSphere 4.x 功能完全实践化的书籍，集成了 VMware 产品、服务器、存储设备、网络设备、软件、系统、操作系统的精华，让你能从头开始创建一个真正的 vSphere 4.x 环境，云计算图书从未如此接近实战！

本书光盘包含海量视频操作，以及完整的操作环境。

本书适合所有对虚拟化技术、云计算感兴趣的读者，尤其适合目前正在实施云计算的公司技术人员。

图书在版编目(CIP)数据

虚拟智慧：VMware vSphere 运维实录 / 胡嘉玺 编著. —北京：清华大学出版社，2011.1

ISBN 978-7-302-24421-9

Ⅰ.①虚… Ⅱ. ①胡… Ⅲ. ①虚拟处理机—应用软件，VMware vSphere Ⅳ. ①TP338

中国版本图书馆 CIP 数据核字(2010)第 258263 号

责任编辑：栾大成
装帧设计：
责任校对：徐俊伟
责任印制：

出版发行：清华大学出版社　　**地　址：**北京清华大学学研大厦 A 座
http://www.tup.com.cn　　**邮　编：**100084
社 总 机：010-62770175　　**邮　购：**010-62786544
投稿与读者服务：010-62795954，jsjjc@tup.tsinghua.edu.cn
质量反馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
印 刷 者：
装 订 者：
经　　销：全国新华书店
开　　本：188×260　**印　张：**43　**字　数：**1180 千字
附 DVD 2 张
版　　次：2011 年 1 月第 1 版　　**印　次：**2011 年 1 月第 1 次印刷
印　　数：1~5000
定　　价：79.00 元

产品编号：040401-01

序

虚拟化是当今最炙手可热的 IT 技术之一。尤其在全球金融危机之后，越来越多的企业把关注的目光投向了虚拟化技术，希望能够利用虚拟化技术减少企业的 IT 成本，降低运维开销以及提高信息安全水平。毫不夸张地说，虚拟化技术是 IT 技术中为数不多的既叫好又叫座的角色。

虚拟化技术如此重要，IT 技术人员自然要重点跟进。可惜目前的 IT 图书市场上缺少虚拟化方面的重量级图书，尤其是缺少系统介绍 VMware vSphere 的图书。这使得很多对虚拟化技术感兴趣的技术人员无法轻易找到适合自己学习的参考资料，只能徘徊在虚拟化技术的边缘。

但是！从现在起，这种缺憾终于不复存在了！胡嘉玺先生编著的这本虚拟化巨著全面系统地介绍了 VMware vSphere 在企业的具体应用。从 VMware vSphere 的版本选购到存储、网络、硬件、软件的具体规划设计。让 IT 技术人员可以利用 VMware vSphere，从无到有地搭建起一个虚拟化平台。本书还对虚拟化导入给出了非常完整的评估步骤，非常具有项目实践的参考价值。

如果您是虚拟化技术的爱好者，如果您希望利用虚拟化技术改善企业的 IT 运维水平，如果您希望了解最好的虚拟化产品有哪些特点，如果您希望自学成为虚拟化高手……您只需要拥有这本书就足够了。这不是广告，我是认真的！

岳雷（虚拟化 MVP）

http://yuelei.blog.51cto.com/

前　言

21 世纪一下子就走完了前十年，IT 产业在这十年也有了等比级数的成长，然而在科技快速进步的同时，软硬件的复杂性也不断提高，让 IT 从业人员需要面对的挑战比从前更难上加难。

但是在 20 世纪的最后一年，两个麻省理工学院的先知，早就看到十年后的 IT 产业，将会是一个所有运作都基于“虚幻”的平台，将硬件抽象化的结果，就是把硬件也使用电子的方式提交、克隆、生成、消灭。以这个思路为基础的虚拟化，正成为 21 世纪下一个十年的重要基石。

VMware 在 2009 年 4 月推出跨时代的 vSphere 4.0，成功地落实了 VI3 的改朝换代，但更具代表性的意义是，这是全世界第一个云端操作系统，并且真正将硬件的 CPU、内存、存储、网络设备等设备抽象化，成为一个可计量的单位，从而将这个计量单位转换成服务性质，让 IT 人员更方便地设计自身企业的平台架构，或是做成新的产品服务，从 Amazon 的 EC2，大家真正发现了虚拟化对硬件的分解能力，自此一股云端的潮流就锐不可挡了！

如果只把 VMware 想成虚拟化产品的公司，那么你很可能将会和这个技术生成极大的鸿沟。将计算机硬件抽象化的观念，不但正在改变旧有的 IT 产业架构，其更不断创造新的 IT 前台，我们耳熟能详的云计算，如果没有虚拟化，怎么可能出现在你我的生活中呢？

本书斗胆敢说是全球第一本将所有 vSphere 4.0 功能完全实践的书籍，集成了 VMware 产品、服务器、存储设备、网络设备、软件、系统、操作系统的精华，让你能从头开始创建一个真正的 vSphere 4.0 环境。截至当前为止，当前全球还没有任何一本书能包含这么完整的 vSphere 功能。写作时虽然辛苦，但能在虚拟化这个领域提供自身在业界多年的心得、获取的成就感，也感到十分值得，以将近一整年的时间落实将近 118 万字的巨作，希望能给读者们带来实质的帮助。

全书之落实，首先要感谢清华大学出版社，虽然整本书比预计推出的时间晚了一些，但整个创作团队对质量的坚持是可以确定的。此外还要特别谢谢黄秀玲小姐在著作期间对我的帮助，以及 VMware、QNAP、IBM、EMC、Dell、HP、Cisco 的工程师不断在技术上的指导。

虚拟技术是一个又广又深的学问，任何一个人都不可能在短时间之内熟练，因此本书不管在技术上或是文本上谬误之处可能在所难免误，也请读者们能不吝指正赐教。读者们如果对书中内容或是对虚拟机技术有任何疑问或意见，请至作者网站留言。

虚拟先生 http://vaemon.com

胡嘉玺

目　录

第 1 章 企业虚拟化的目的及本质

关键词：

- 什么是虚拟化
- 虚拟化的历史及现状
- 虚拟化有哪些种类
- 企业虚拟化的场合及特色
- 虚拟化和云计算的完美结合
- 信息从业人员的挑战

这一阵子“云计算”、“虚拟化”、Amazon EC2、VMware 这些字眼充斥在各式各样的介质上，好像技术不和这些字眼沾点边，就不是最新的趋势了。其实虚拟化这个技术早就出现在你我的生活中，而“云端”、EC2 这些新的字词，更是和虚拟化脱不了关系，我们在开宗明义的第 1 章，就来看看这些已经让人困惑的技术词汇，对你我的生活到底有什么影响。

1.1 什么是虚拟化?

这本书的读者对象大部分是计算机玩家，但如果你问“什么是虚拟化”，我想大部分人的回答都会是“就是在一个操作系统中运行另一个操作系统”。虽然这个答案也没错，但这并不是真正“虚拟化”的意义，只能说是虚拟化在硬件和操作系统之间的一个实践。那到底什么是虚拟化呢？我们下面就来看看在计算机不同层次上，虚拟化的真正定义。

1.1.1 定义计算机的服务层级

计算机的服务层级一般是由下图所示构成，从最底层的硬件开始，向上有操作系统、软件运行的框架（或称为 Framework）、应用程序、应用程序提供的服务以及最上层的用户等。任何信息系统大都遵循此架构，也是计算机系统几十年来一直不变的架构。



▲ 计算机的服务层级一般由这几个部分组成

1. 硬件部分

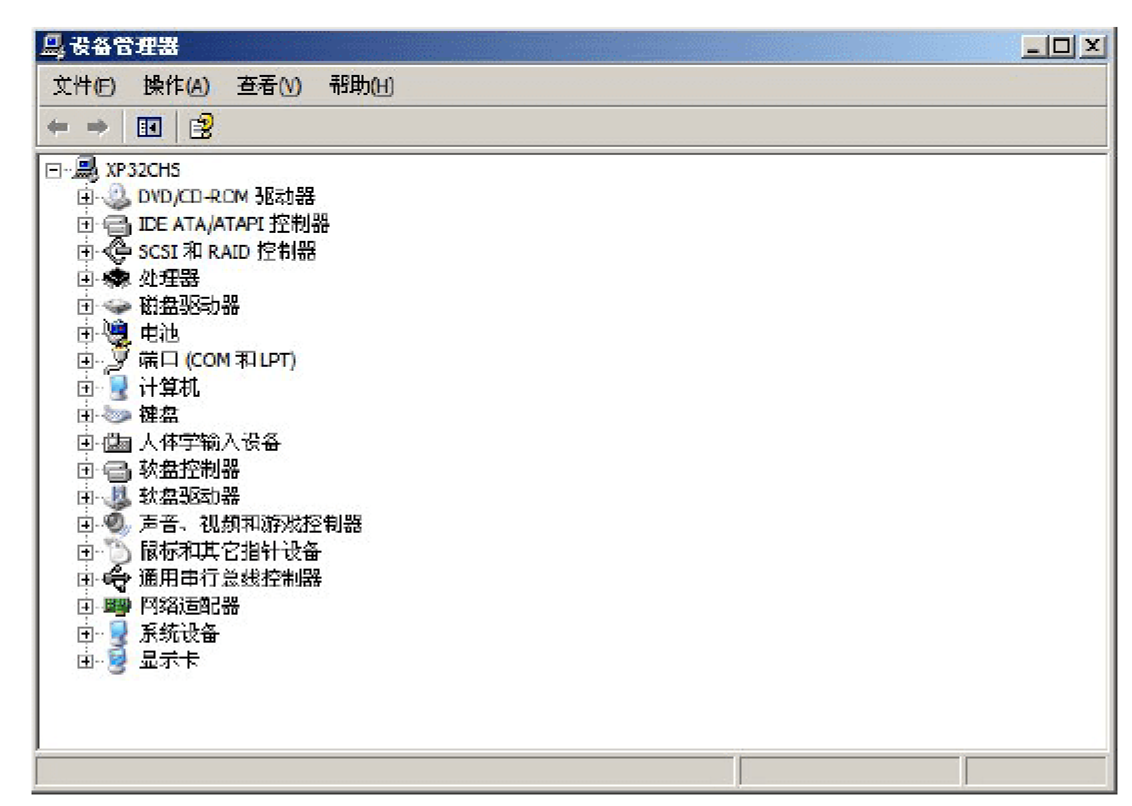
硬件厂家虽然可以用各式各样的新科技来制作先进的产品，但还是得考虑到产品的通用性。以 CPU 为例，虽然各种 CPU 厂家都以高速低耗电为主要设计原则，但以信息业来说，还是有几个必须遵守的架构，如 Intel 架构、PowerPC 架构等。这也是硬件厂家在设计时的较少制约。



▲ 以虚拟化来说，当前的 CPU 架构以 Intel 的 X86 为主

2. 操作系统部分

操作系统的功能很复杂，与本书有关的部分，主要还是硬件与上层的沟通。举例来说，如果你买了一片新的显示适配器想要玩三维游戏，必须先安装驱动程序才能发挥硬件的功能及效能。这时操作系统的用处，就是提供游戏和硬件之间沟通的管道（驱动程序），因此没有操作系统的话，硬件和用户之间是被隔离的。



▲ 操作系统提供的功能有很大一部分是为了和下层硬件进行沟通

3. 框架部分

大家都有使用 IE 的经验，如果你在使用 IE 时，只将“C:\Program Files\Internet Explorer\iexplore.exe”克隆出来，再拿到另一台电脑使用，这个 IE 是无法运行的。原因是这个 IE 在运行时，虽然有运行文件了，但还需要底层的框架提供各种功能。这些框架就是所谓的底层架构（Framework）。这么做的好处是让程序开发人员有一个共通的平台，并且也能确保开发出来的软件能在任何安装 Framework 的计算机上运行。Java Runtime、Microsoft Framework 就是常见的例子。



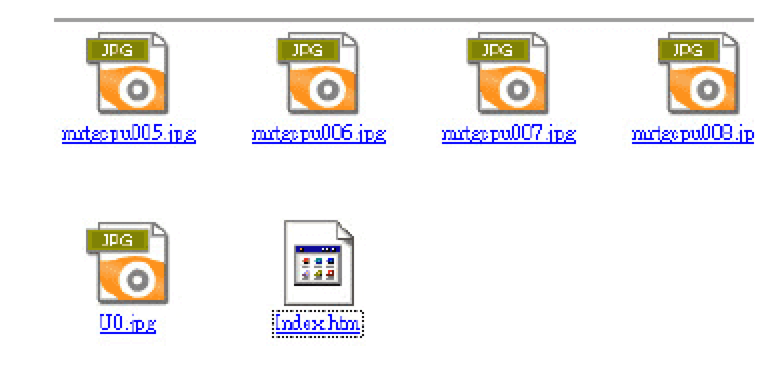
▲ 没有 Framework 大部分的软件是无法运行的

4. 软件部分

软件就是我们看到单独的应用程序，如 MSN、Word 等。当我们要使用软件时，只要运行该软件的运行文件就可以。计算机中软件的单位都是运行文件，再大的软件都有一个代表性的运行文件。而网页上的软件，则由 index.html 这一类的首页来给定，或是由 Web Server 来给定软件的入口点。



▲ 本地的软件是以文件为入口点



▲ 网络的应用程序一般是以一个网页文件为入口点

5. 服务部分

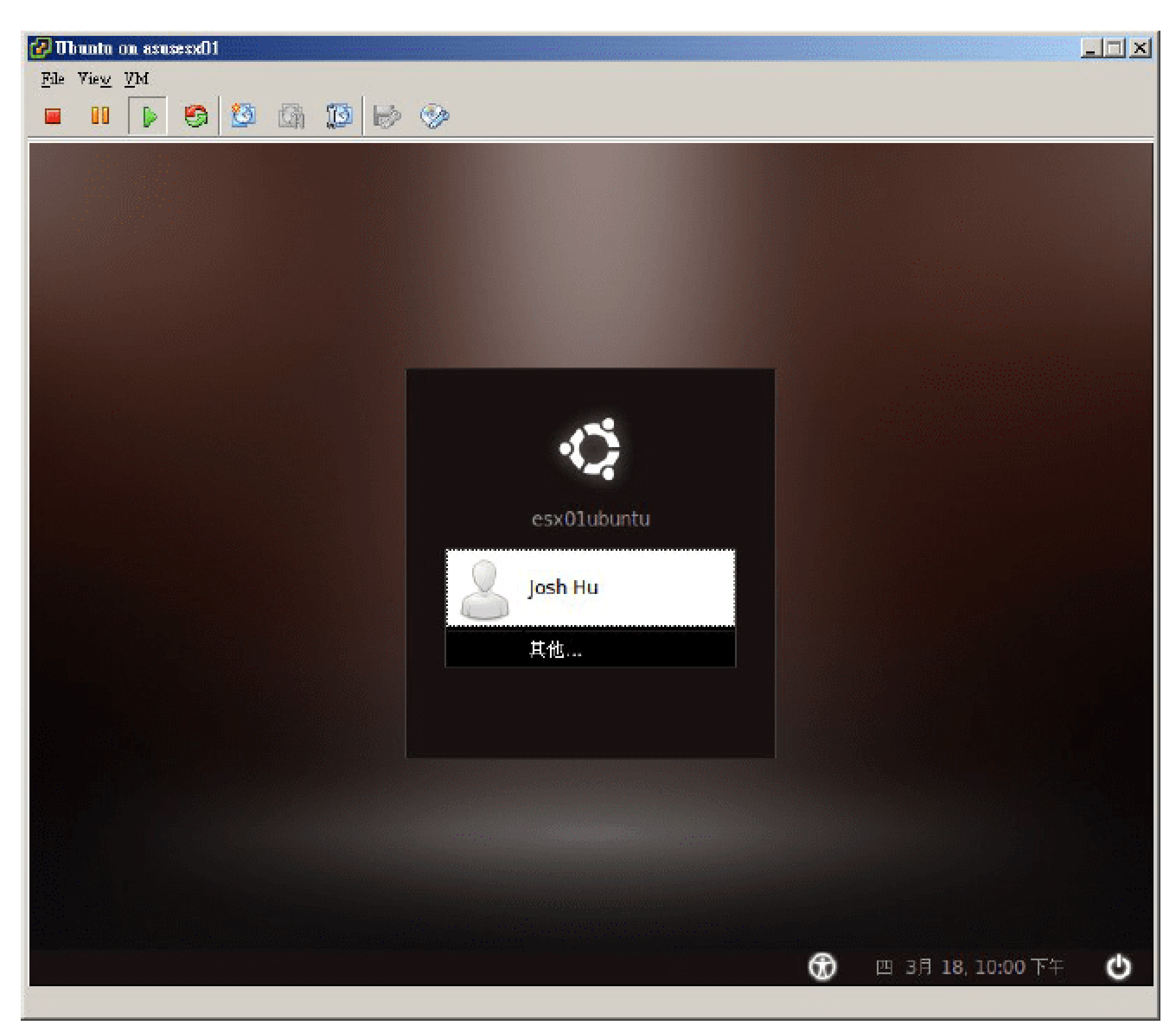
软件呈现出来的功能称为服务。一般来说，一个现代的软件服务包括了物理数据（放在数据库系统中）、业务逻辑以及界面（Interface）。用户通过界面，以业务逻辑为工具来操作物理数据，就是一个基本的服务模式。



▲ 软件提供的功能称为服务，如银行安全控件就是一个实时通服务

1.1.2 虚拟化可以打破层级依赖的现状

事实上，这些不同的层级之间与当前的架构是紧紧依赖的。没有软件的话，服务就无法提供给用户；没有 Framework，软件就无法运行；没有操作系统的话，就无法安装各式各样的软件和 Framework；没有硬件当然就什么都没有了。为了避免层次之间的紧密依赖性，在 1960 年代，就有人引入虚拟化的概念，做法很简单，就是将上一层对下一层的依赖撤销；换句话说，就是将本层的依赖从底层中抽离出来，因此我们定义“虚拟化”的正规说法，可以为“虚拟化，就是不断抽离依赖的过程”。

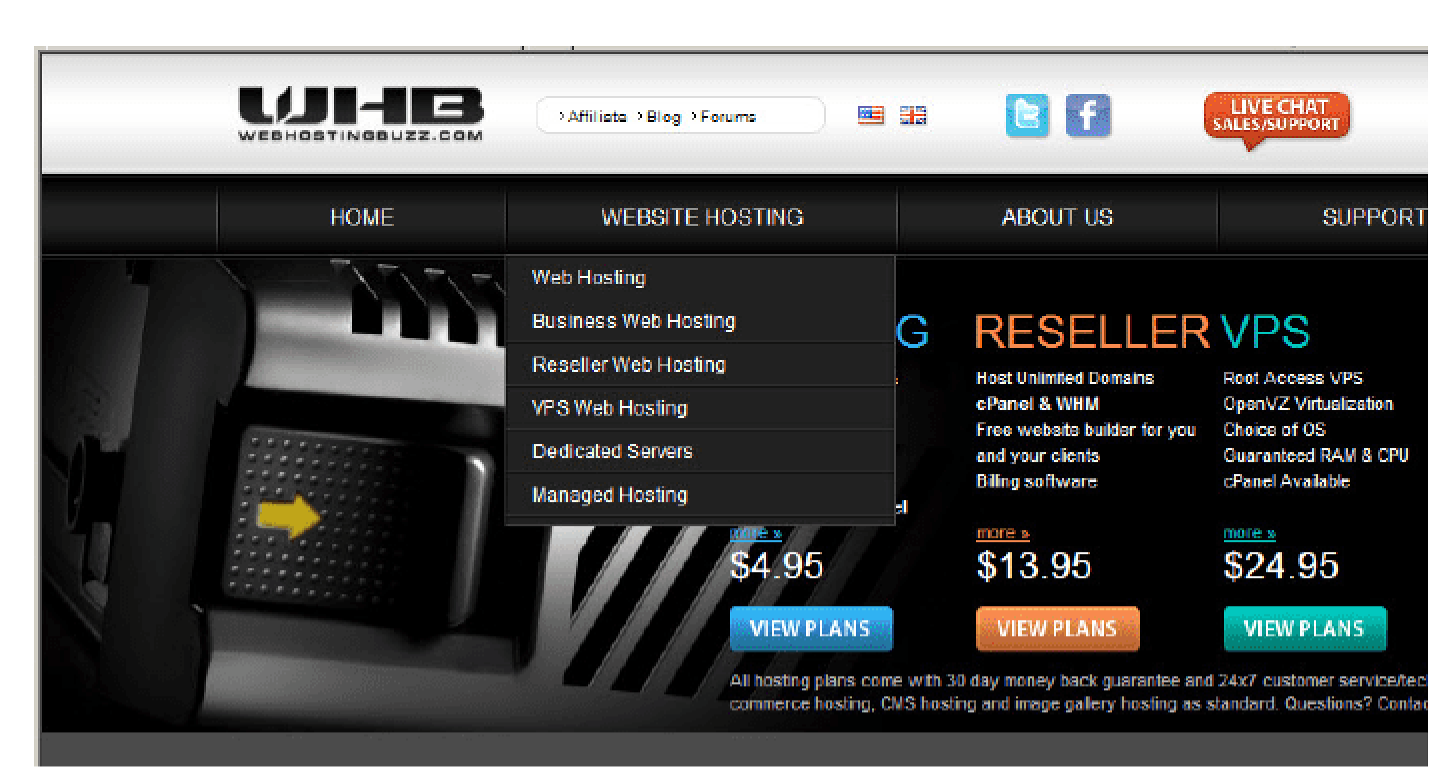


▲ 虚拟化的操作系统就是打破和硬件的依赖性

理解	“虚拟”从字面上看就是“假”的，意味着“本来没有这个东西，但要假装让你觉得有，以达到我们使用的目的”。事实上，这个较白话的解释，就是当前虚拟化的真正实践原则。

1. 服务虚拟化的例子

我们就举“虚拟主机”这个例子。通常在申请网站时，需要一个域名和对应的 IP，但 IP 不够，因此我们可以利用 Web Server 中的配置，让多个域名指向一个 IP。按照前面的解释，就是“让域名能脱离对 IP 的依赖”，而另一个解释更清楚，就是“原来没有这么多 IP 来一对一指向域名，我们就假装有这么多 IP 对到不同的域名”，因此一个 IP 可以对多个域名，节省 IP 的目的就达成了。



▲ 虚拟主机就是服务虚拟化的最好例子

2. 软件虚拟化的例子

最常见的就是可携式软件（或称绿色软件，Portable Software）了。有些软件放在 USB 随身盘中，带到哪里都可以运行，这种软件和下层 Framework 的依赖被打破，不需要 Framework 也可以运行。



▲ 可携式软件就是将和 Framework 的依赖打破

3. Framework 虚拟化的例子

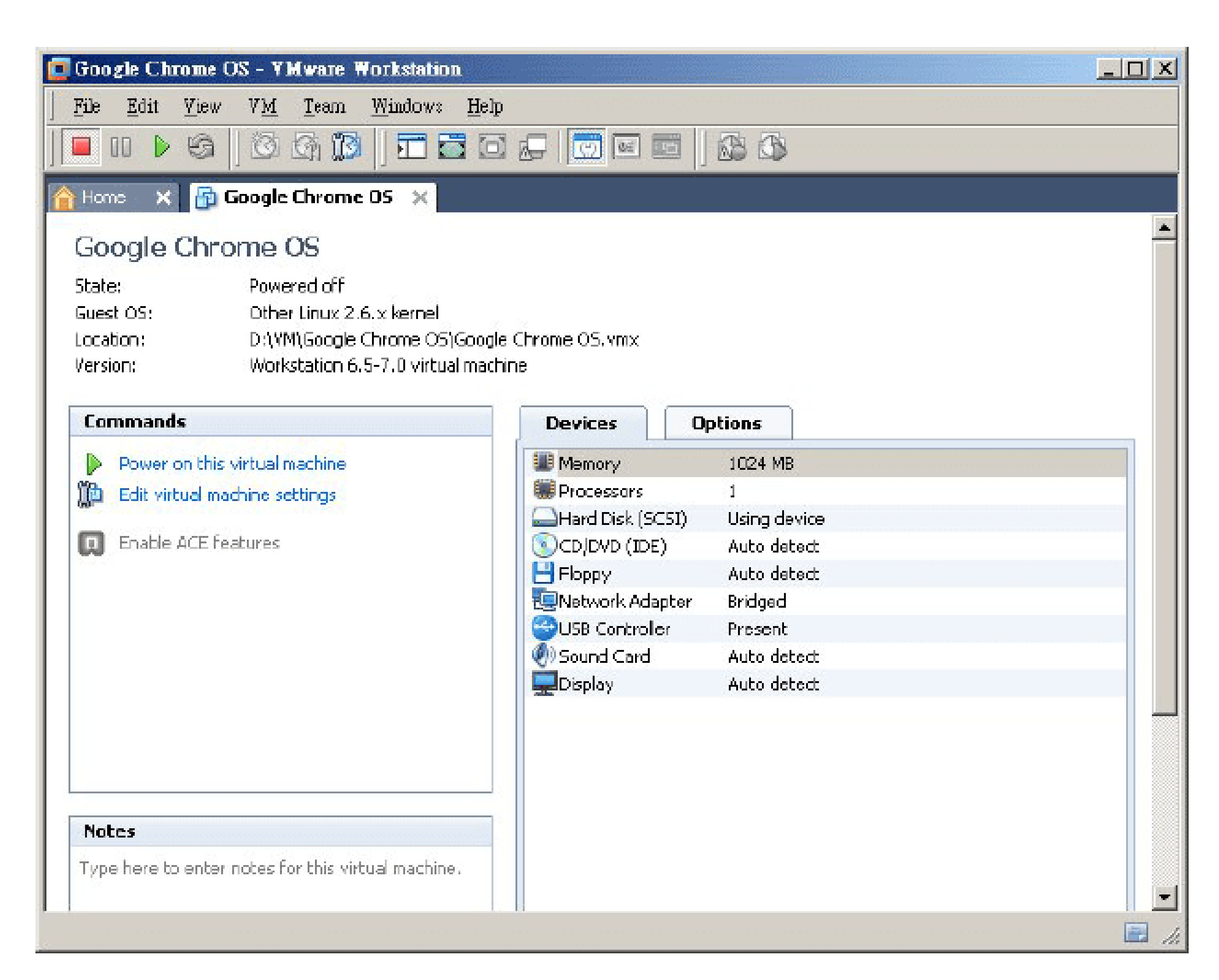
让 Framework 不再受制于操作系统，让这个 Framework 支持的应用软件都能运行在各式各样的操作系统之上。当前做得最好的应该就是 Java Runtime。虽然在不同的操作系统上都要安装不同版本的 Java，但不同的操作系统都能运行 Java 的 Runtime 算是一个较贴近的例子。



▲ Java 的 Runtime 可在任何 OS 上运行

4．操作系统虚拟化的例子

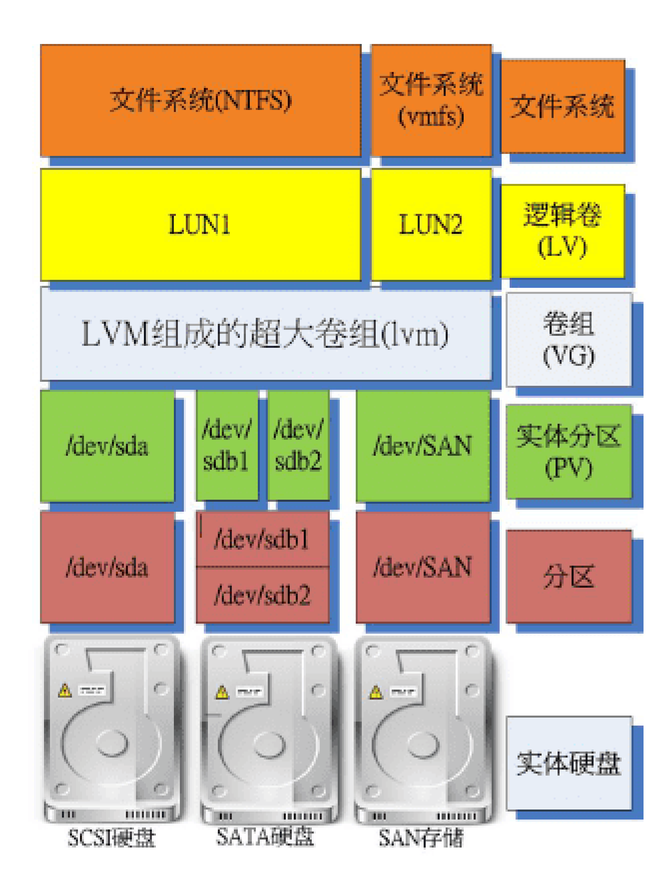
这就是我们本书的热点了！让操作系统不再依赖硬件，直接可以运行在一个统一的“硬件界面”上。本书所介绍的产品 VMware vSphere 就是最好的例子，vSphere 提供了一个“硬件界面”，让一台服务器上能并发运行多个操作系统，让操作系统都以为“自身在一台物理机器上”。本书所提到的“虚拟化”，在没有特别给定时，都是专指操作系统虚拟化（或称平台虚拟化）。



▲ VMware 就是标准的 OS 虚拟化（或称平台虚拟化）的最好实例

5．硬件还能虚拟化？

硬件还能虚拟化吗？当然可以。最好的例子就是存储设备的虚拟化。我们可以将多个硬件组合成一个大存储池，并且依照我们的需要将这个存储池再分割。在本书稍后会有专门的章节来介绍存储设备。



▲ 存储设备的逻辑卷（Logical Volume）观念就是最好的虚拟化实例

1.2 平台虚拟化的历史

虚拟化可以说是当代的显学，举例来说，当前全球第三大软件厂家，就是专门生产虚拟化产品的 VMware。除了专门的虚拟化产品之外，云计算、SaaS、存储设备等都和虚拟化紧紧相关。但在 2010 年的今天，虚拟化的历史可以追溯到 1960 年代，溯古知今，我们就来看看。

1.2.1 虚拟化小史

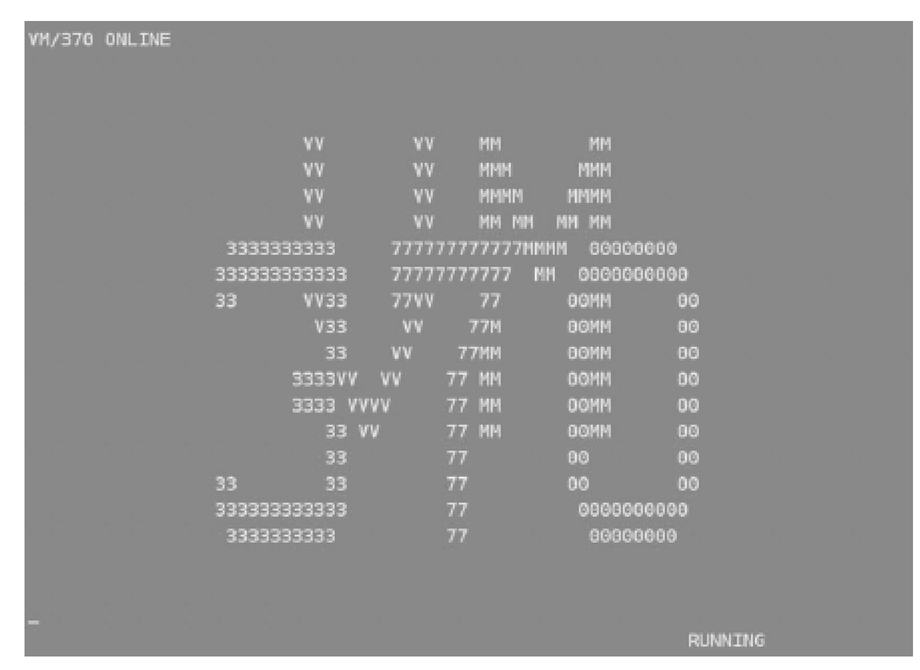
虚拟化不是今天才有的新技术，在 20 世纪 60 年代，IBM 就已经推出了虚拟化的产品。

1. 1960 年：IBM 的 CP-40

在 1960 年，IBM 的 CP-40 就完全展现了“完全虚拟化”的实力。在其上运行了 14 个 IBM S/360 的虚拟机，每一个虚拟机上拥有 256KB 的内存，并且可以对映到不同的硬盘分割区上，更可以使用共享的资源，如打印机，可以并发开发 CP 和 CMS 系统，并且也提供了安全上的功能。



▲ IBM 的 CP-40 是第一台支持虚拟化的主机



▲ 图为 CP-40 运行一个 VM370 的画面

2. 1966 年：IBM 的 S360/Model 67

在 1966 年的 7 月，当时的信息企业龙头 IBM 推出了 S360/M67。这是一台 32 比特内存寻址的机器，可以插上多个 CPU，并且支持了完整的虚拟化功能。换句话说，在 S360/M67 上可以运行多个当时的应用操作系统，并且 IBM 也提供了一个转换层。当时可以运行虚拟化的 S360/M67 称为 CP-67，而这台 CP-67 对后来的任何虚拟化产品都算是启蒙的始祖。



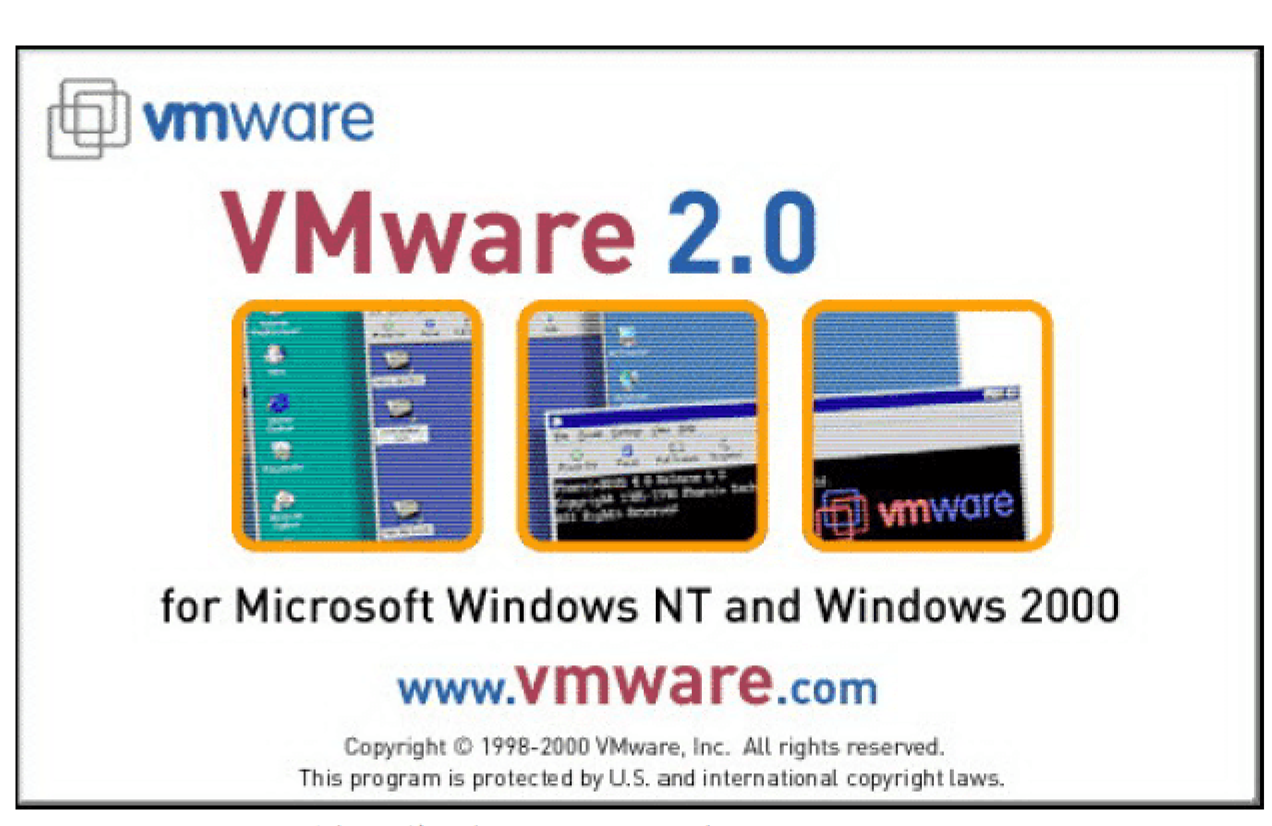
▲ 这是有名的 S360/Model 67

1.2.2 近代的 X86 虚拟机产品发展

在大型计算机时代，虚拟机的使用都在所谓的 MainFrame 上，但随着软、硬件的进步，大多数个人用户使用的 X86 平台上也开始有人动虚拟化的脑筋了。在 20 世纪 90 年代末期，虚拟化也正式进入 Windows 系统，飞入寻常百姓家，成为你我生活的一部分。

1. 1998 年：VMware 的产品

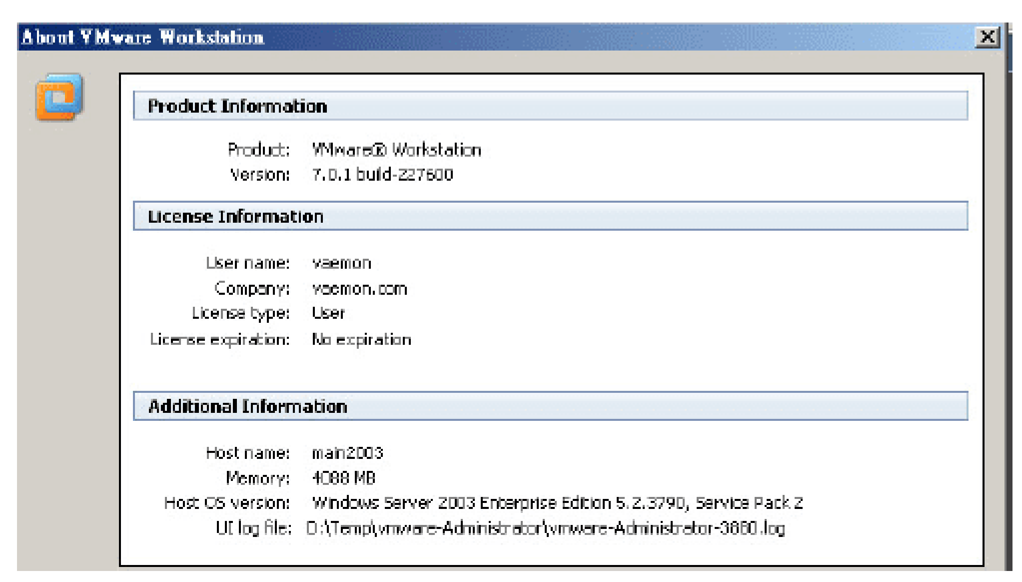
在 2000 年以前，X86 架构上并没有什么虚拟化的产品，而虚拟化在当时也绝非主流的趋势。但早在 1998 年，毕业于 MIT 的 Diane Greene 女士已察觉计算机资源的过低使用率，因此和 Dr. Mendel Rosenblum、Scott Devine、Dr. Edward Wang 以及 Edouard Bugnion 等人成立了 VMware 公司，专精于 OS in OS 的软件，期待能发挥硬件的完全效能，并且也为当时的专业 IT 人员提供一个测试、评估的低成本环境。在 1999 年，VMware 推出了 X86 上的第一款虚拟化商用软件 VMware Workstation。这款 Workstation 可以让当时的 Windows 2000 上运行多个 Windows 系统，Workstation 版本的 VMware，也让这个名字在信息业打响名号，为现在的 VMware 打下良好的基础。



▲ 最早期的 VMware 产品：VMware 2.0

2. 2000 年之后：百家争鸣

眼看着 VMware 做大，各大厂家也开始重视虚拟化的市场。首先还是一些小型的软件厂家开始发难，如 Windows 系统下的 Virtual PC、Parallels 的 Workstation 以及 VirtualBox 等。这些小型厂家最后都变成了大公司的试金石，相继被微软、Sun 或是 IBM 并购。时至今日，虚拟机已经成为各大软件公司的兵家必争之地，甚至连硬件厂家在推出产品时都要全力支持虚拟化的命令，本书稍后会有详细的介绍。



▲ 当前 VMware Workstation 的版本也到 7 了

1.3 虚拟化的种类及简介

简单来说，平台虚拟化就是“OSes in an OS”，让一台物理计算机能并发运行多个 OS。在平台虚拟化中，最重要的功能，就是让其上运行的多个 OS 觉得自身好像拥有独立的机器，而不是和别的 OS 分享。

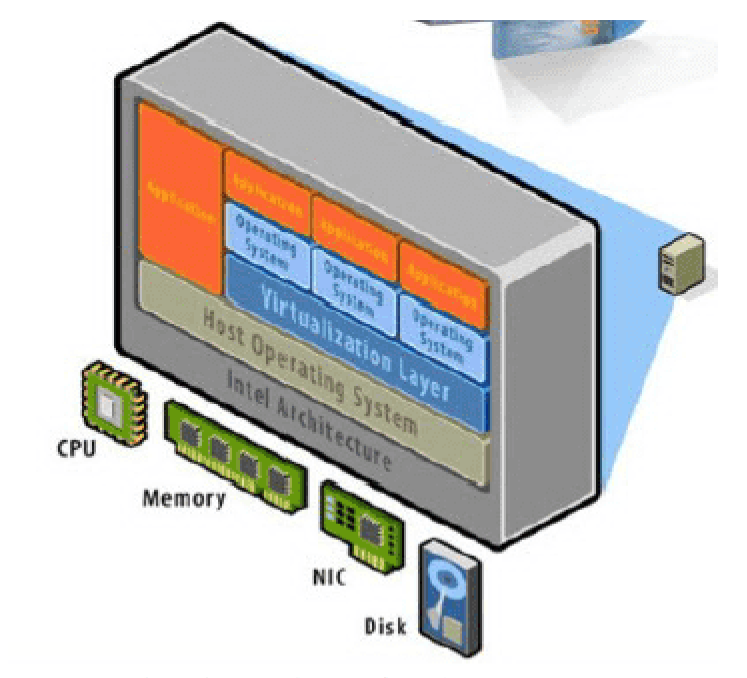
提供这个功能的机制，通常被称为虚拟机管理员（Virtual Machine Monitor，VMM），更常用的名称为 Hypervisor。Hypervisor 提供的功能又要多，又不允许和 VM 中的 OS 抢资源，因此提供一个好的 Hypervisor 是虚拟机厂家的最高机密，也是不同虚拟机产品的差别，我们就从 Hypervisor 的观点来看不同种类的虚拟化。

1.3.1 原生及寄居架构

以是否存在宿主操作系统来看，虚拟机的 Hypervisor 主要分为原生及寄居两种架构。所谓原生架构，就是 Hypervisor 本身不需要依赖任何操作系统，或是说 Hypervisor 本身就是一个操作系统，只是这个操作系统的目的是专门服务于虚拟化。而寄居架构，则是 Hypervisor 被看成一个应用软件或是服务，运行在已经安装好的操作系统上才能运行。

1. 寄居架构

这一类产品最大的特色就是必须在一个已经安装好的操作系统上进行，称为寄居架构（Hosted）。这么做的好处就是硬件的兼容性。只要寄居的操作系统能使用的硬件，虚拟机中的操作系统都能使用到。然而这种运行在桌面操作系统上的 Hypervisor，被视为寄居操作系统上的一个应用软件，虽然在安装时会将不少 Hypervisor 的部件放入内核，但在寄居的操作系统出现任何问题时，VM 中的操作系统将无法使用，无法满足重视安全及稳定的企业应用上。



▲ 图为标准的寄居架构，从中可以看到 VM 寄居在一个操作系统上

举例来说，最有名的虚拟产品 VMware Workstation 或是微软的 Virtual PC/Server，都必须安装在一个操作系统之上，如 Windows 7 或 Windows Server 2003。如果这个操作系统蓝屏了，其上运作的 VM 客户端当然就全部一起蓝屏。这就是寄居架构。

▶ **寄居架构的主流产品**

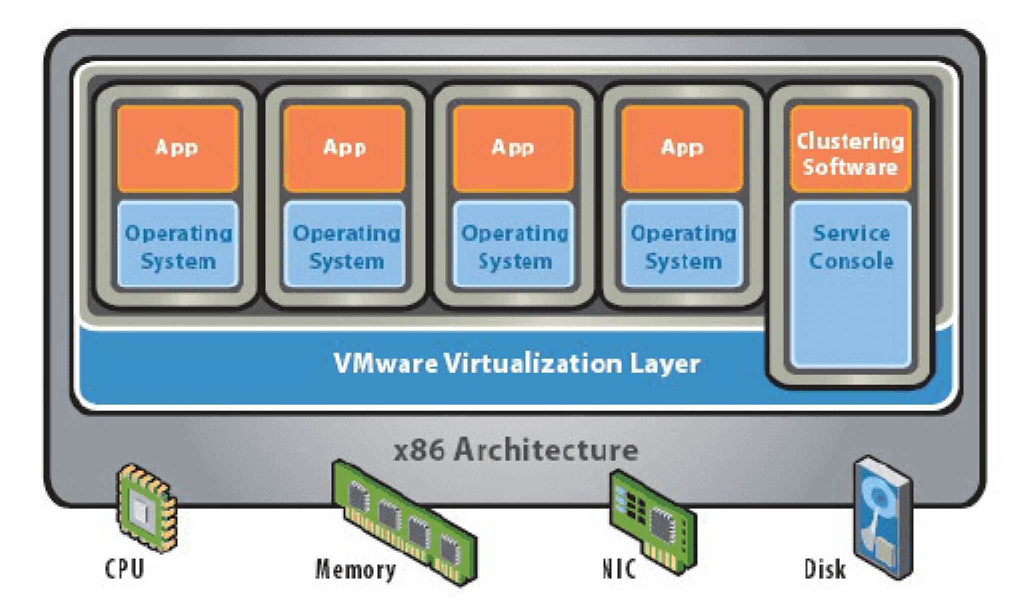
- VMware Workstation（有 Windows 及 Linux 版本）
- VMware Server（有 Windows 及 Linux 版本，免费）
- VMware Player（有 Windows 及 Linux 版本，免费）
- 微软 Virtual PC 2007
- 微软 Virtual Server 2005 SP2（免费）
- 微软 Windows Virtual PC（只能在 Windows 7 上运行，免费）
- Parallels Workstation
- SUN Virtual Box（已被 Oracle 收购）



▲ VMware 的 Player 也是很流行的免费寄居架构产品

2. 原生架构

相对于寄居架构的就是原生架构，原生架构上，Hypervisor 直接安装在硬件上，将所有的硬件资源接管。由于 Hypervisor 层极小（可小到 32MB footprint），并且不管理太复杂的事项，仅负责和上层的 VM 操作系统沟通及资源协调，蓝屏的概率更小。在其上的任何一个操作系统蓝屏了，都不会影响其他的客户端，因此较适合于企业应用。

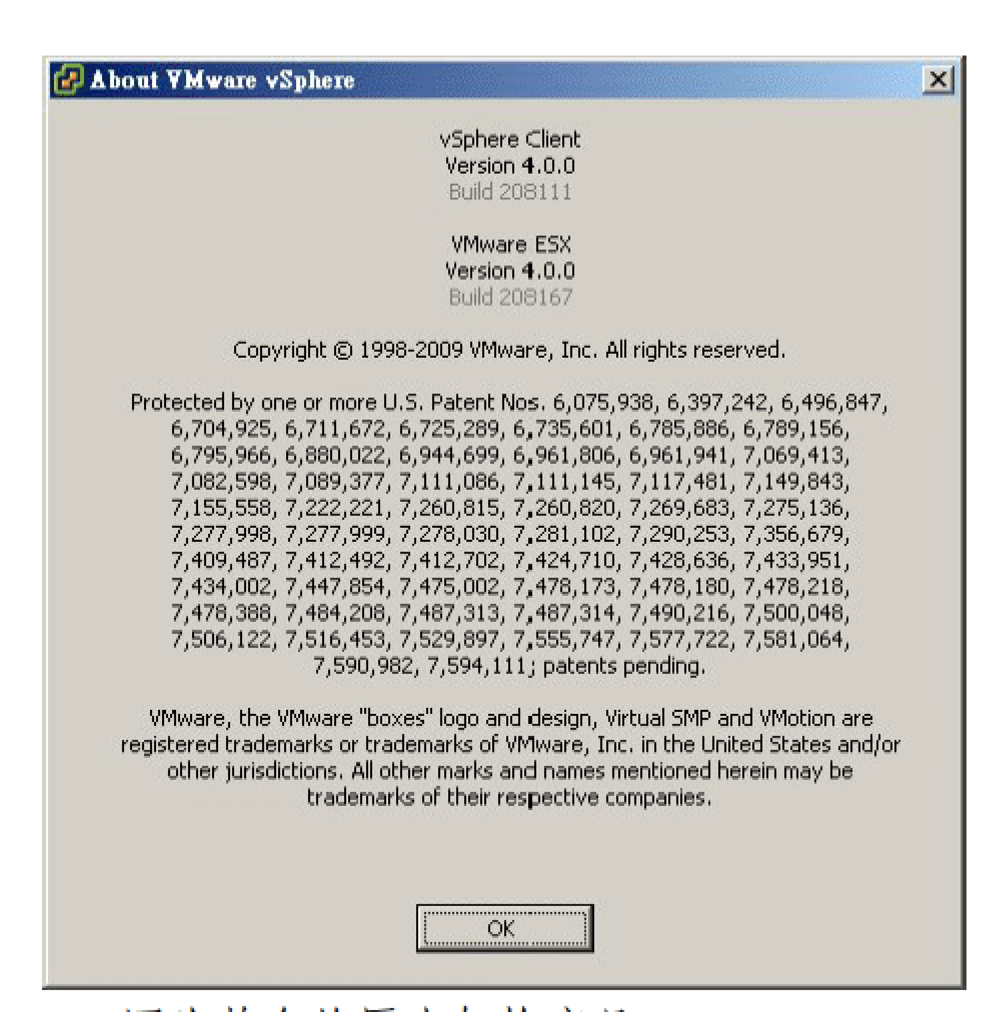


▲ 原生架构不需要操作系统，Hypervisor 直接管理硬件

当然原生架构的虚拟机为了保持其稳定性及微内核，自然不可能将所有桌面产品的驱动程序都放入，因此最大的问题就是硬件兼容性。大部分的原生架构产品都支持主流服务器及存储设备，如 NAS、iSCSI 或 FC SAN，并且强调这些设备都经过了官方测试，为的就是确保在企业运行环境中的稳定性。一般 PC 所使用的硬件，大部分无法在原生架构的虚拟机下运行。

▶ **原生架构的主流产品**

- VMware vSphere
- 微软 Hyper-V 2008 R2（免费）
- 微软 Windows 2008 R2 Hyper-V Role
- Citrix XenServer/XenDesktop
- Xen（Linux，开源）
- Qemn（Linux，开源）



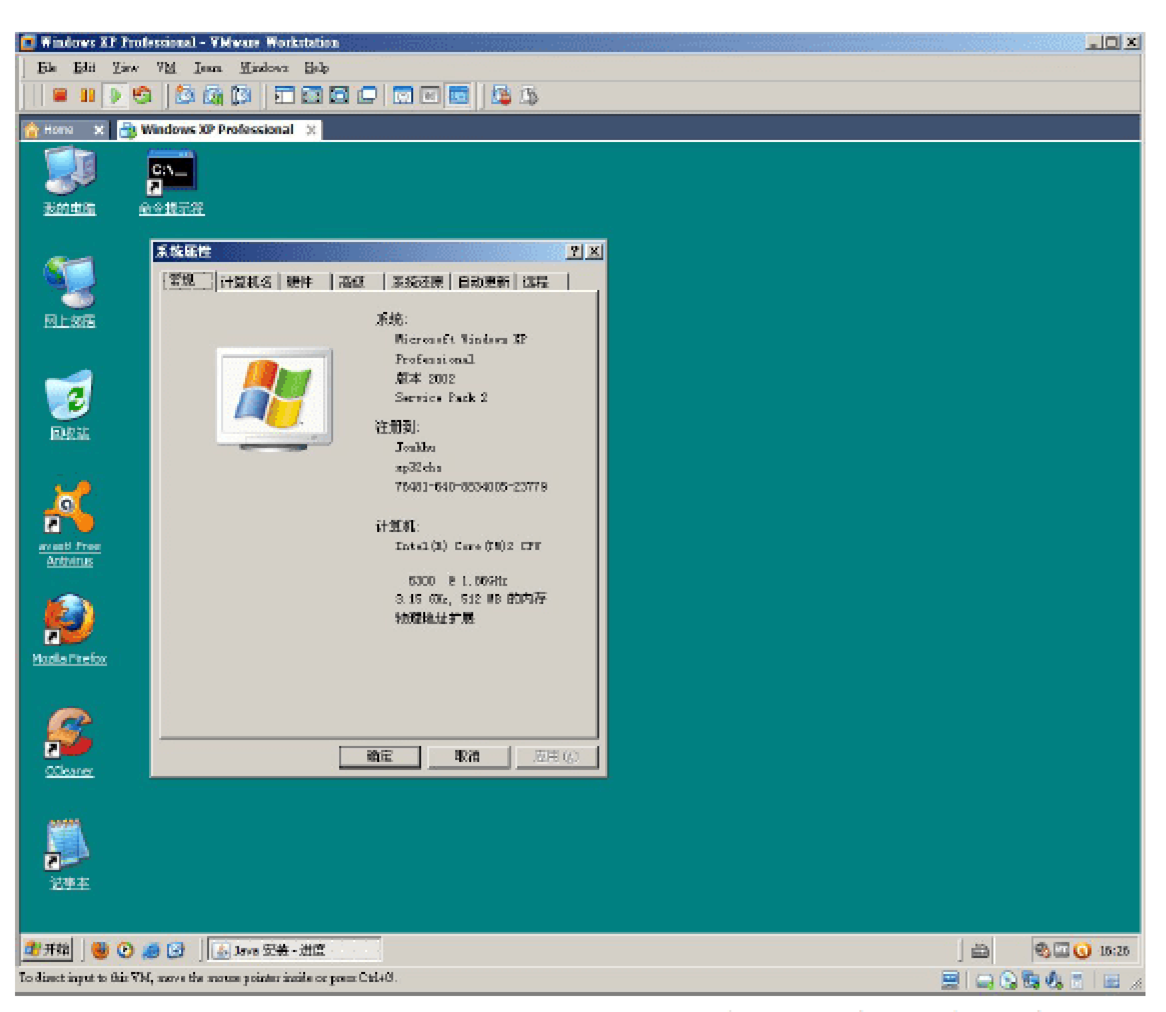
▲ 图为著名的原生架构产品 VMware vSphere

1.3.2 以虚拟化的程度区分

在 Hypervisor 本身，针对虚拟化程度的不同，也有不同的产品类别。

1. 完全虚拟化（Full virtualization）

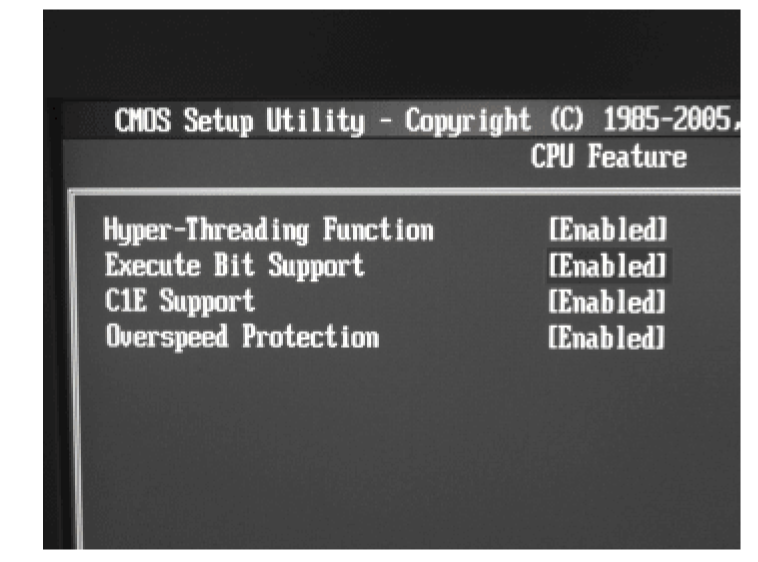
当前大部分的主流虚拟机产品均属此类，主要是客户端操作系统在这种虚拟产品上运行时，不需要做任何修改。如 VMware Workstation、vSphere、微软的 Hypervisor 等。



▲ 操作系统在运行时，根本不觉得自身在虚拟机中

2. 硬件辅助虚拟化（Hardware-assisted virtualization）

或称硬件虚拟机（HVM），主要是操作系统在其上运行时，必须靠系统的硬件来落实虚拟化的过程。在 Intel 的 VT 和 AMD 的 AMD-V 推出之后，主流虚拟机厂家也在现有产品的基础上，推出新版本来融入此功能。当前所谓的硬件辅助，通常都是指虚拟化中的特有功能，如 Intel 的 VT-d 功能，在不支持的服务器上，虽然能安装 vSphere，但却无法使用特定的功能，在本书稍后有完整的说明。



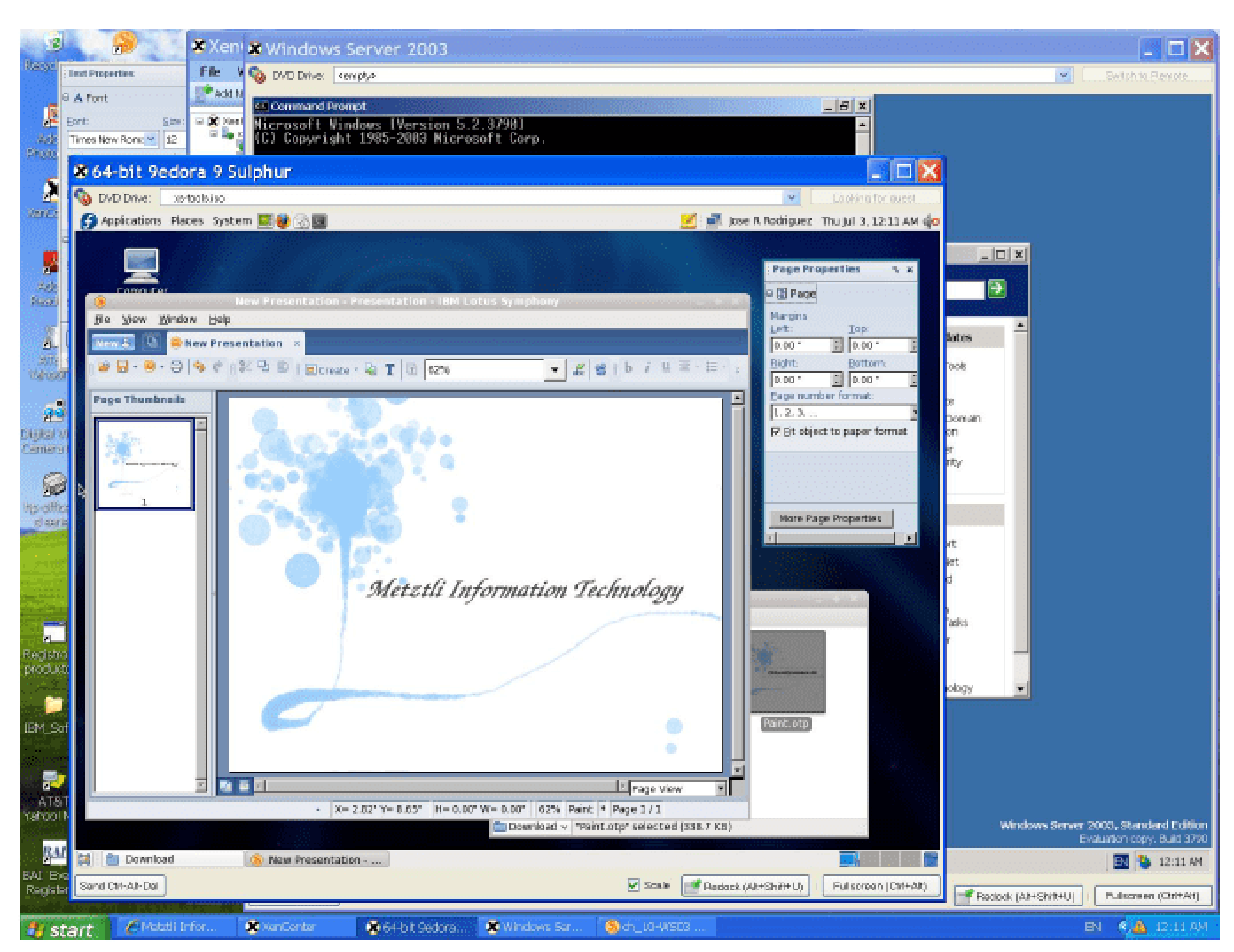
▲ 大部分的 CPU 当前都已经支持虚拟化的硬件辅助了

3. 部分虚拟化（Partial virtualization）

此类虚拟机产品在运行时，系统只提供部分的虚拟化，无法常规安装完整的操作系统，在早期化产品的版本中还常见到，但当前在 X86 平台下，已经很少有这样的产品了。

4. 半虚拟化（Para-virtualization）

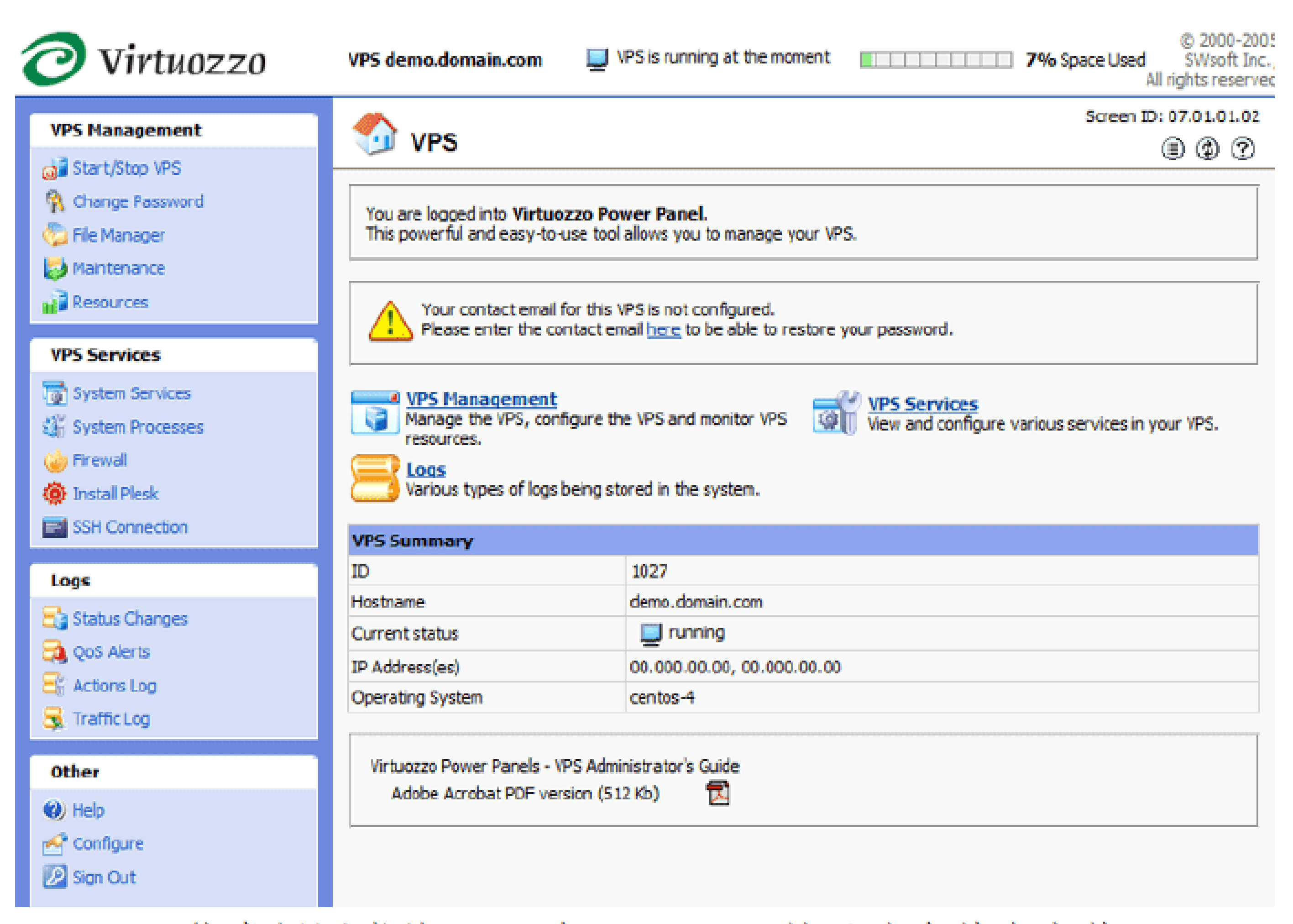
或称“半虚拟化”或“泛虚拟化”。操作系统运行此类产品时，必须修改内核才能顺利安装，早期的 Xen 就属于此类，现在也较少有类似的产品。



▲ 早期的 Xen 属于此类，现在 Xen 也全虚拟化了

5. 操作系统级虚拟化（OS-level virtualization）

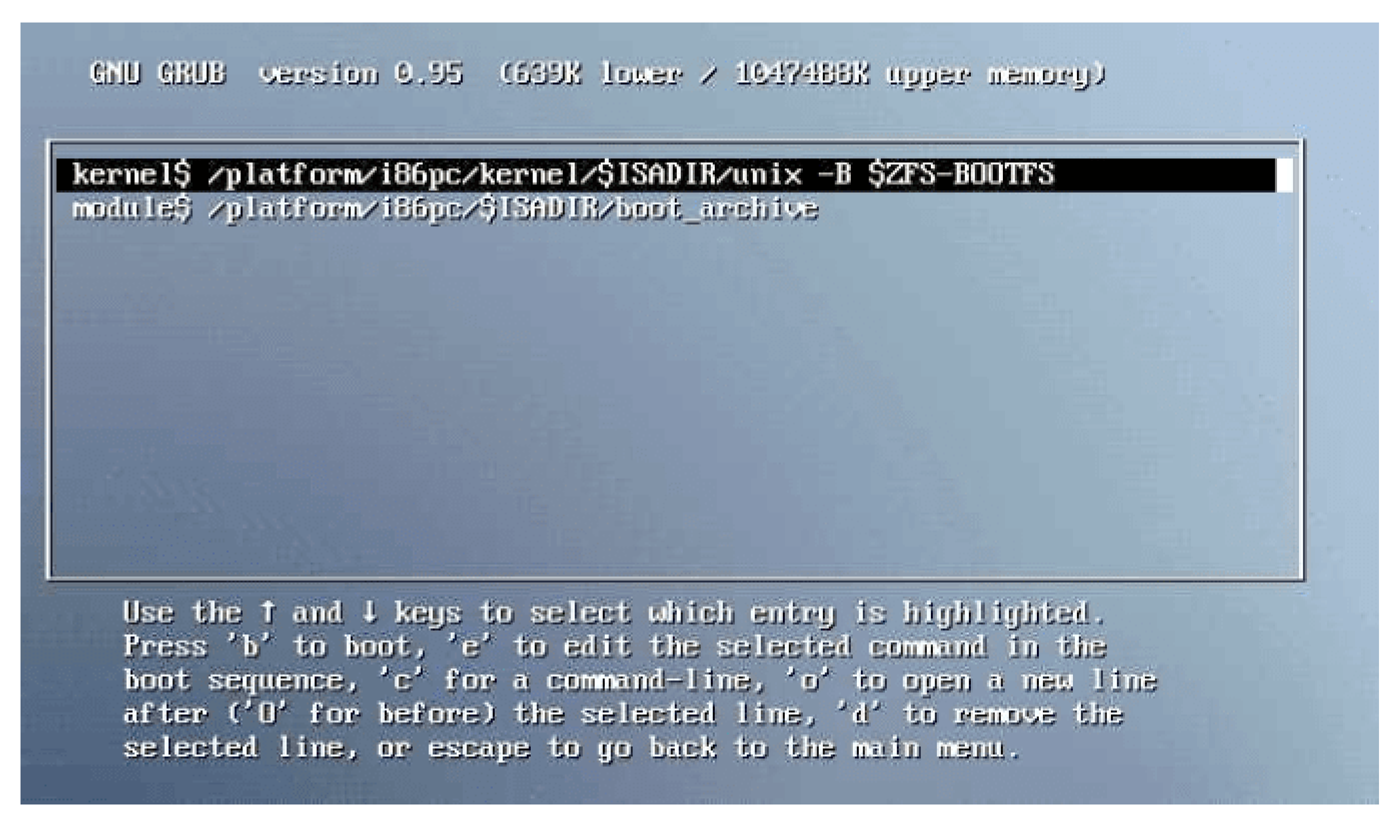
这一类的虚拟机产品只能类比出主机的系统，如 Linux 上只能运行 Linux，Windows 上只能运行 Windows，最有名的就是 Virtuozzo，是当前大部分国外数据中心厂家在提供的 Virtual Machine Host 的主流技术。



▲ 只能虚拟同类的 OS，如 Windows 就无法在其上安装 Linux

6. 寄居环境（Hosted Environment）

最常见的就是 Linux 下的 User Mode，可以在一个 Linux 下，以应用软件的方式来运行其他的 Linux。注意这和 Hosed 寄居架构的完全虚拟化是不同的。



▲ Linux 下的寄居环境是最有名的寄居环境

1.4 企业虚拟化的场合及目的

虚拟化的起因很简单，就是因为硬件资源的浪费，主要针对的问题就是硬件资源效率的低落。在计算机 CPU 和内存的效能和数量以穆尔定律倍数成长的同时，CPU 和内存在操作系统中的使用效率低落的情况反而加重。所谓的效率低落，就是无法完全发挥 CPU 的完整性能。虽然软件和操作系统的专家不断改良效率，但速度远远比不上 CPU 和内存发展的速度，因此让单个硬件平台运行多个操作系统的观念，成为解决这个问题的最好答案。当前大部分服务器的 CPU 使用率常在 5%以下，内存更在 30%以下，因此把多个操作系统放在一台机器中，多少可以让 CPU 的利用率高一些。



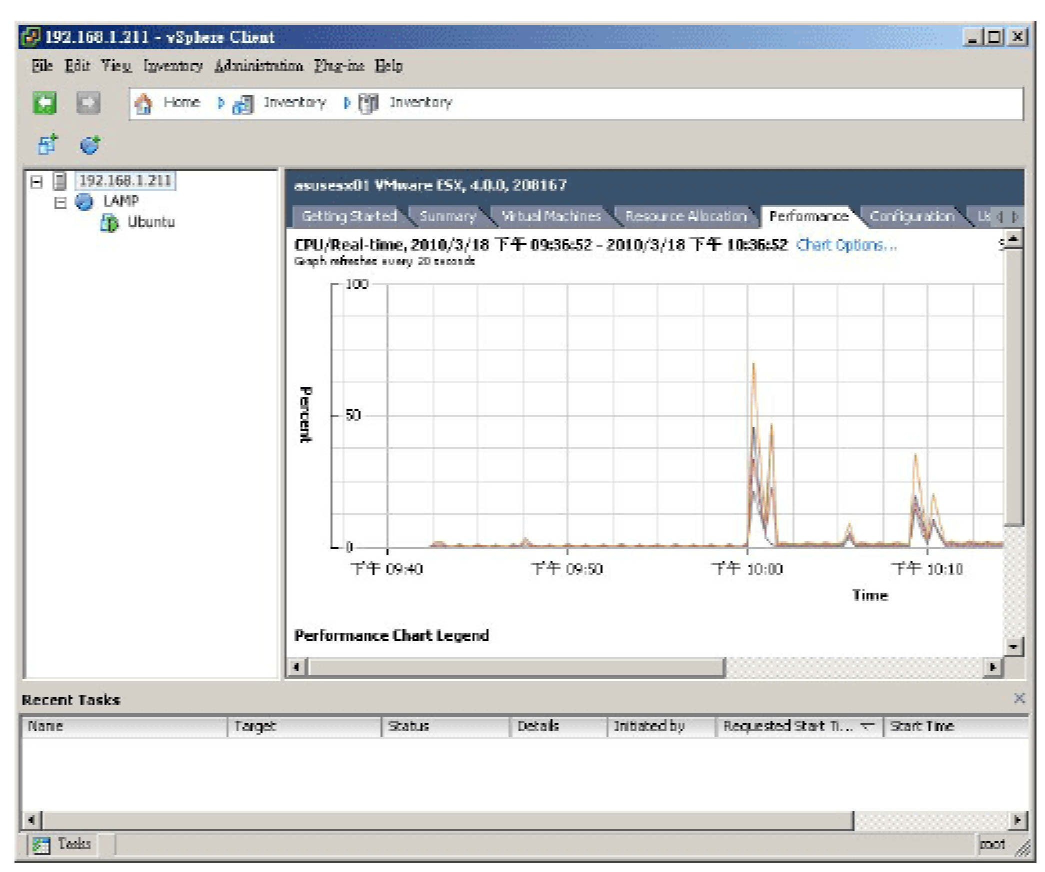
▲ 虚拟化的起因很大一部分原因是因为服务器的 CPU 使用率太低

1.4.1 企业虚拟化的场合

前面提到将多台 OS 放到一个服务器上可以加强硬件资源的使用率，因此服务器合并（Server Consolidation）是最常见的虚拟化企业应用。既然可以合并服务器，那是否能合并桌上计算机呢？当然行，因此服务器合并和虚拟桌面架构（Virtual Desktop Infrastructure）是当前企业虚拟化的两大热点，而随着这两大热点所生成的，就是最夯的云计算了。

1. 服务器合并

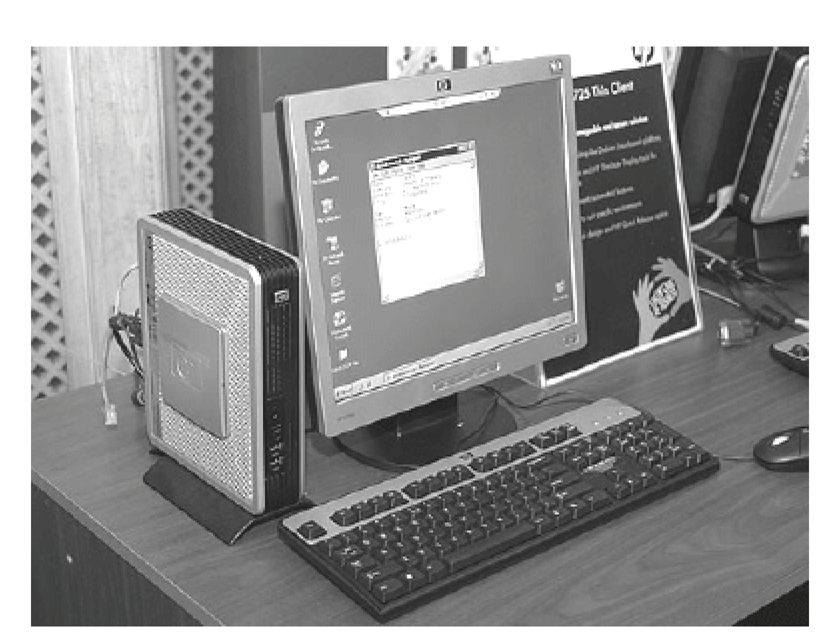
不管是在企业内部还是提供主机托管的数据中心，当前已经很少有单个主机使用单个 OS 了。大部分的服务器早已合并，使用的就是虚拟化技术。服务器合并的好处多的说不完，本书介绍的 vSphere 就是个中翘楚，而服务器合并之后，最明显的优点就是可减少 IT 初期成本（Capital Cost）的支出以及电费冷却的运营支出（Operational Cost）。但这只是其中的一小部分。



▲ 多台服务器在一台物理服务器上运行，可以让 CPU 的效率大一点

2. 企业桌面环境管理

当前虚拟机在这方面的应用上，就是将客户端的桌面操作系统全部移到服务器上的虚拟机。每一个桌面用户都连入自身的虚拟机（使用 RDP 或是特殊的联机软件，如 Citrix 的 ICA）。这么做的好处除了可以省下一大笔升级硬件的预算，更可以将所有的桌面操作系统集中管理，不管是升级、安装应用程序、用户权力管理等，都可以大量简化 IT 的管理成本。



▲ 把桌面计算机都虚拟化之后，客户端可以用规约型计算机（Thin Client）

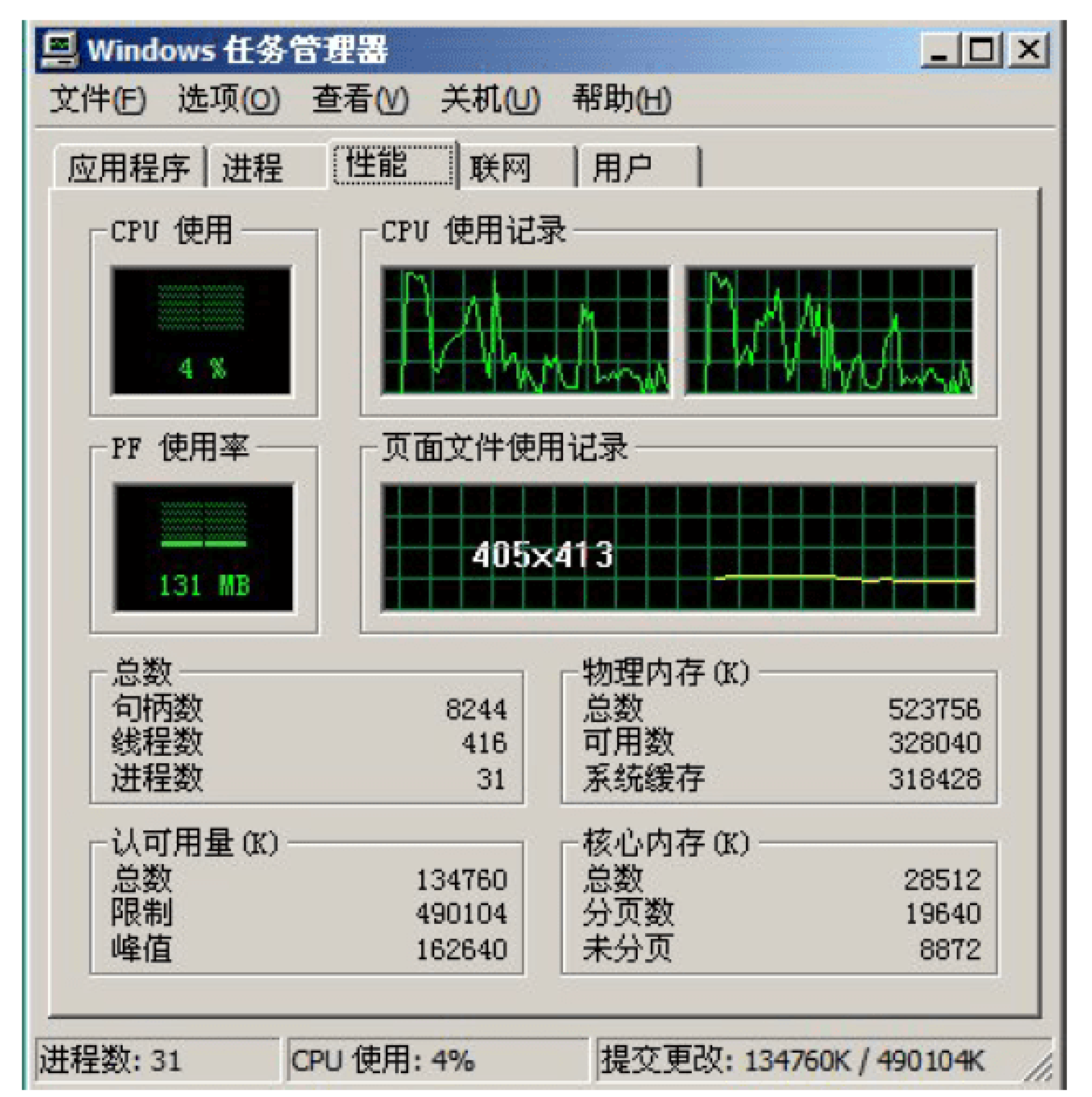
1.4.2 企业虚拟化的优势

虚拟化并不是将服务器合并而已，也不是光能省下电费和买机器的钱，在当今这么复杂的 IT 环境中，虚拟化的设计不断考验着公司的 CIO 及 IT 人员，因此要设计一个完善的企业基础架构，匆忙导入或使用虚拟化是十分不明智的，需在这之前必须考虑到更多的细节。

1. 提高硬件资源效率

随着网络环境的过度膨胀，加上服务器的空间、耗电、散热成本不断提高，CPU 等资源利用率过低，使得虚拟机厂家开始将目标放在“单个物理服务器上运行多个操作系统环境”。这样可

以让每一个系统服务（如数据库、网页服务器）在单个的操作系统上运行，而多个操作系统可以在同一台物理服务器上并行运行，不但保持了服务隔离，更让前面提到的所有问题迎刃而解。



▲ 一台多内核的计算机，CPU 竟然只有用到一点点，不是很浪费吗？

2. 管理的优势

虚拟化提供的功能可以将 IT 管理人员再度合并。一个人管理上千台服务器不再是梦想，不但可以让机器有高效性，公司的人事也可以有高效性，当然完善的管理功能必须创建在良好的架构之上。



▲ 整个机房只需要一个人管理，相信吗？

虚拟化提供的另一功能是可以将服务隔离。企业用 IT 环境和桌面应用本质上就有极大差异，在讲究安全、稳定、高可用性和便捷管理的企业环境来说，“服务隔离（Isolation）”是一个基本的原则。举例来说，将 Web 服务器和 SQL 安装在同一个操作系统中就是十分不明智的选择，因为两个服务之间的资源竞争将使两个服务都变慢，而操作系统、Web 或 SQL 任何一方蓝屏将导致整个系统崩溃。因此大部分的企业服务，都强调一个操作系统上只安装一个服务。

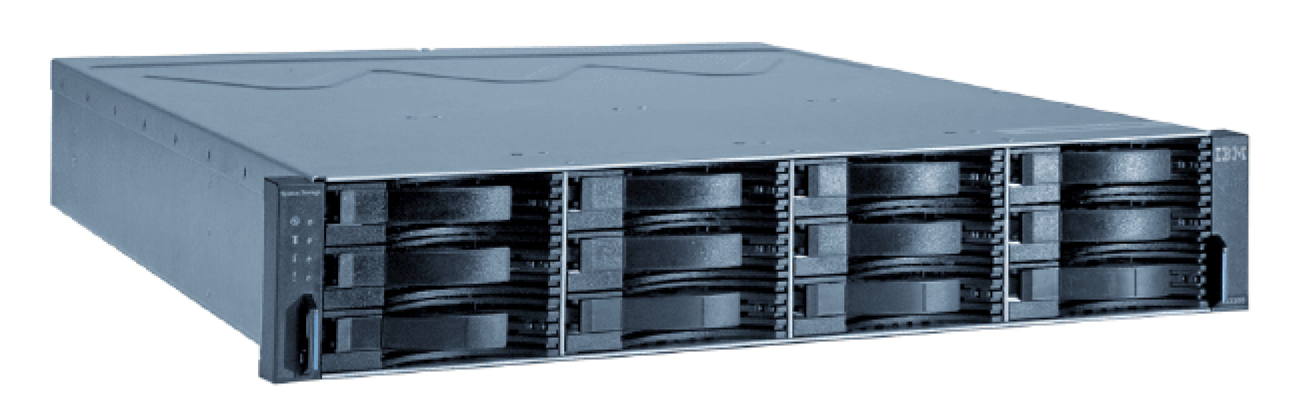


▲ 服务隔离是企业上虚拟化的最大诱因之一

3. 高可用性

在服务器合并之后，大家发现虚拟机的功能不仅于此。由于虚拟机的硬件在抽象化之后，比物理机的应用更有弹性。再加上特殊的硬件和设计之后，企业最在乎的高可用性（High Availability）、冗余、负载均衡、副本等从前必须靠复杂技术或是昂贵设备的问题，使用虚拟化竟然可以一并解决。

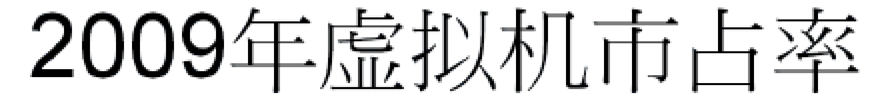
此外，虚拟化更可以解决当前设备无法解决的问题，包括动态主机迁移、快捷删除数据、统一桌面管理，甚至是创建永远不会蓝屏的企业集成环境等，都是新一代虚拟机企业应用的明日之星。

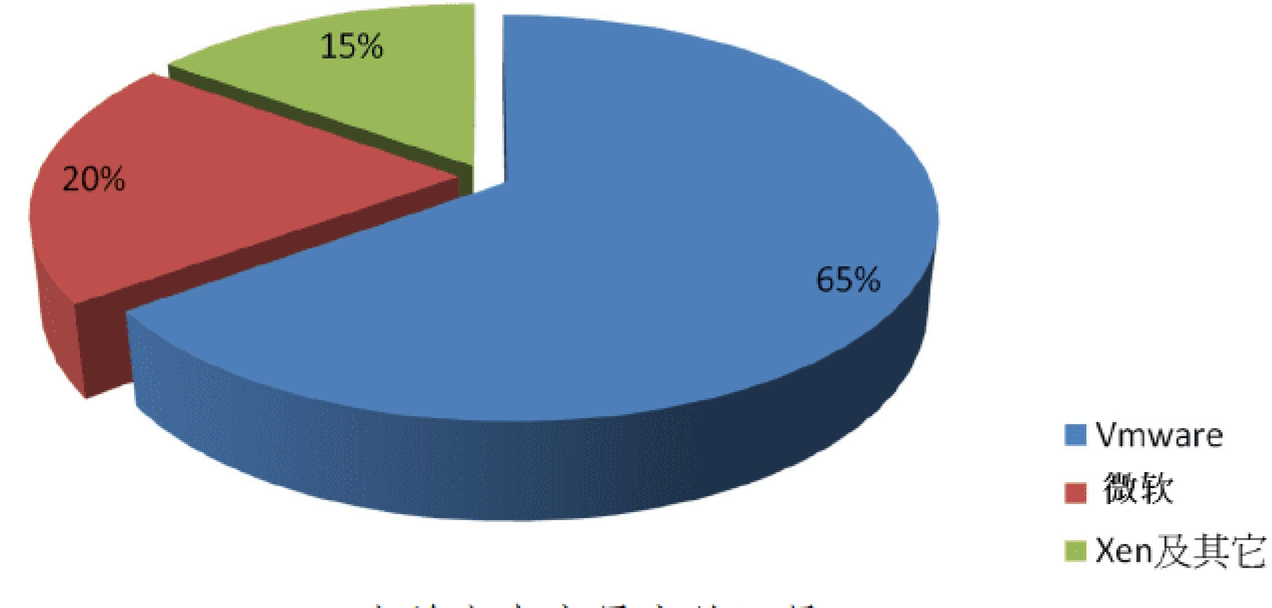


▲ 在搭配了适当的硬件之后，用虚拟化可创建永不蓝屏的环境

1.5 企业虚拟化的主流产品

以市场占有率来说，当前企业虚拟化的主要产品分别是 VMware 的 vSphere、微软的 Hyper-V R2 以及 Citrix 的 XenServer/XenDesktop。当然还有一些小型厂家，但当前其市场规模可忽略不计。下面就来看看这些厂家当前的产品及动态。





▲ 当前市占率最高的还是 VMware

1.5.1　VMware vSphere

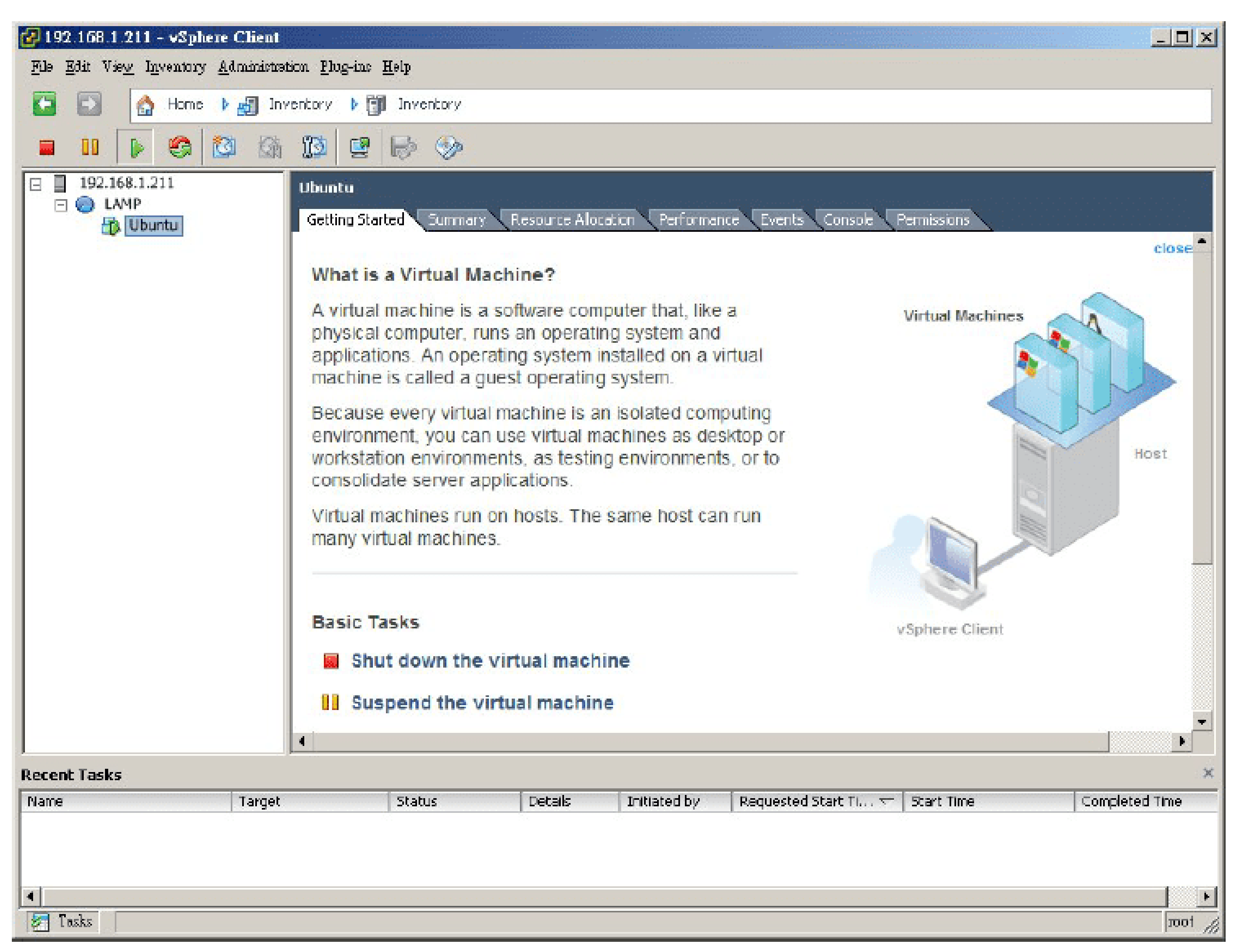
VMware 可说是虚拟界的微软，更是全世界第三大软件公司，Fortune 100 的企业中 100%都使用 VMware 产品，而 Fortune 500 大企业有 98%都使用 VMware，可见其影响力。2009 年 4 月 21 日，在微软紧紧追赶的压力下，VMware 推出了新一代的 vSphere 解决专案。VMware 号称 vSphere 是一个云端操作系统，并且对硬件的支持更加完整，在本书编写时，最新的版本为 4.1。



▲　VMware 是当前全球第三大软件公司

1. 以 ESX 为基础

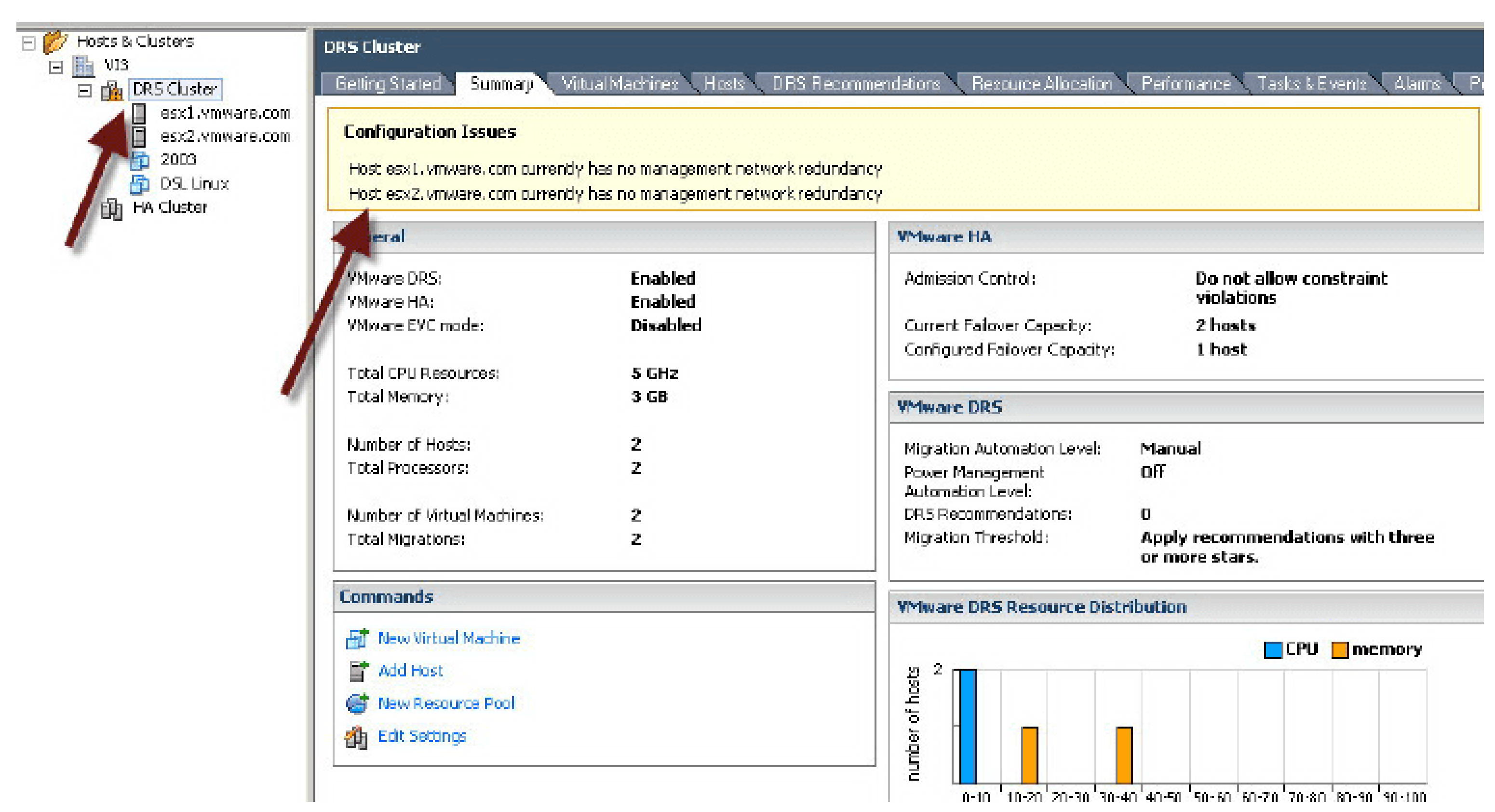
vSphere 以原生架构的 ESX/ESXi Server 为基础，让多台 ESX Server 能并发负担更多个虚拟机。vSphere 不只是一个多台 ESX 的群集，还加上了著名的 VirtualCenter、配合了主流的数据库软件来管理多台 ESX 及虚拟机。



▲　著名的 ESX 服务器

2. 各种不同专用功能

vSphere 最大的特色就是在多台 ESX 加入后，可以落实虚拟机转移的魔法。举例来说，当一台在物理服务器 ESX01 上的虚拟机 VM01 运行过程中，这台 ESX01 突然蓝屏了。由于虚拟机都是硬盘文件放在独立的 SAN 上，那么这个虚拟机所处的 ESX01 只是在内存中的状态消失，但在 SAN 上的虚拟机文件还存在。此时 vSphere 可以将这台在 ESX01 上的 VM01 立即在没有蓝屏的物理服务器 ESX02 激活，用户仅会感觉到小小的断线（或是完全感觉不到）。这类的功能在物理环境是不可能发生的，因为就算做集群，两台机器的配置很可能不一样，也无法保持内存的状态，而这种只有虚拟机才有的特异功能，在 vSphere 上更不止这一个。



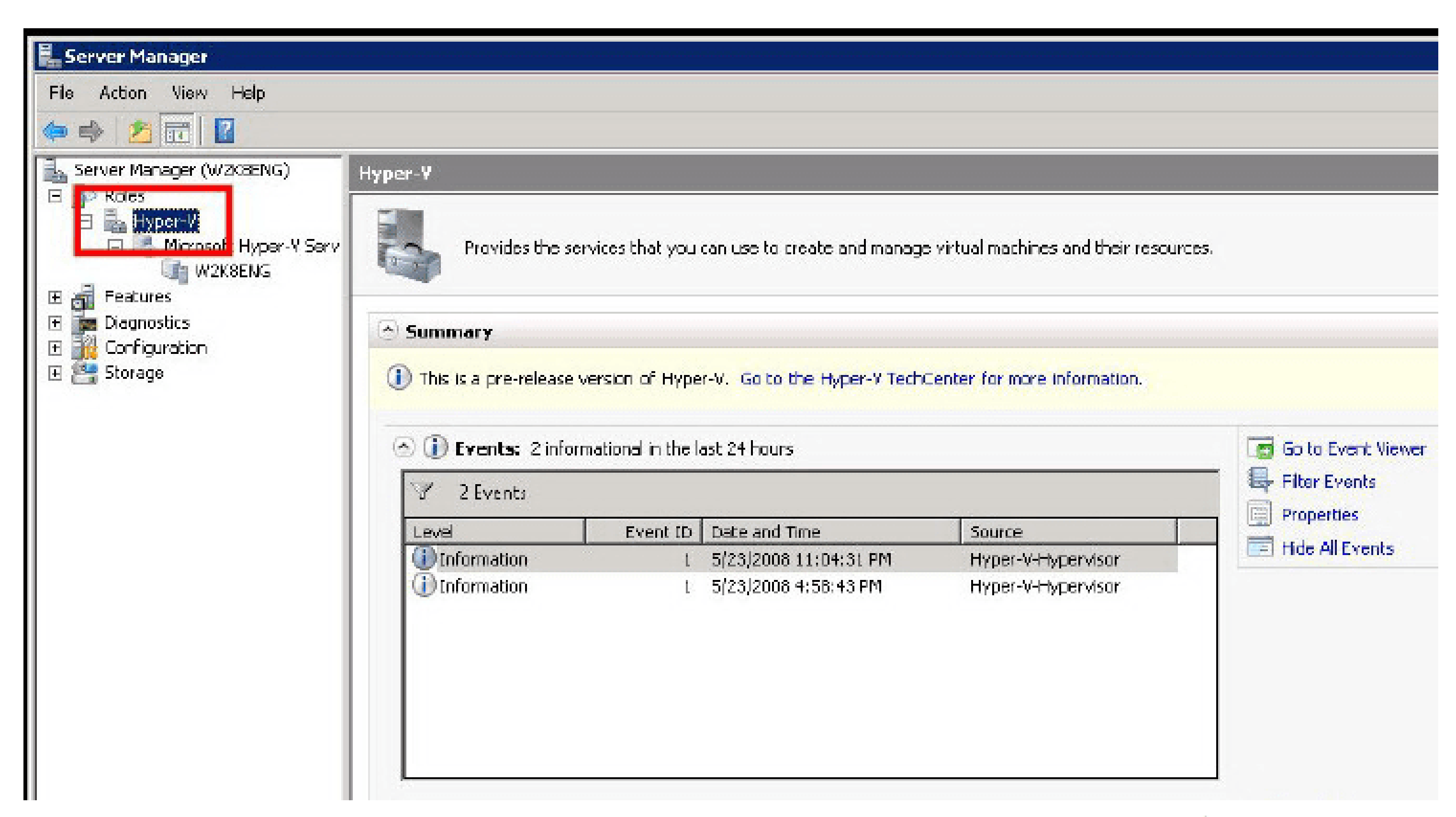
▲ VMware 的 HA 功能可以让 VM 在不同的物理机中跑来跑去

3. 小结

vSphere 的出现改变了人们对虚拟机的看法，更让企业应用从单纯的服务器合并到取代整个企业基础架构，在越来越强调效率的企业环境中，将重写企业 IT 的观念。

1.5.2 微软的 Hyper-V R2

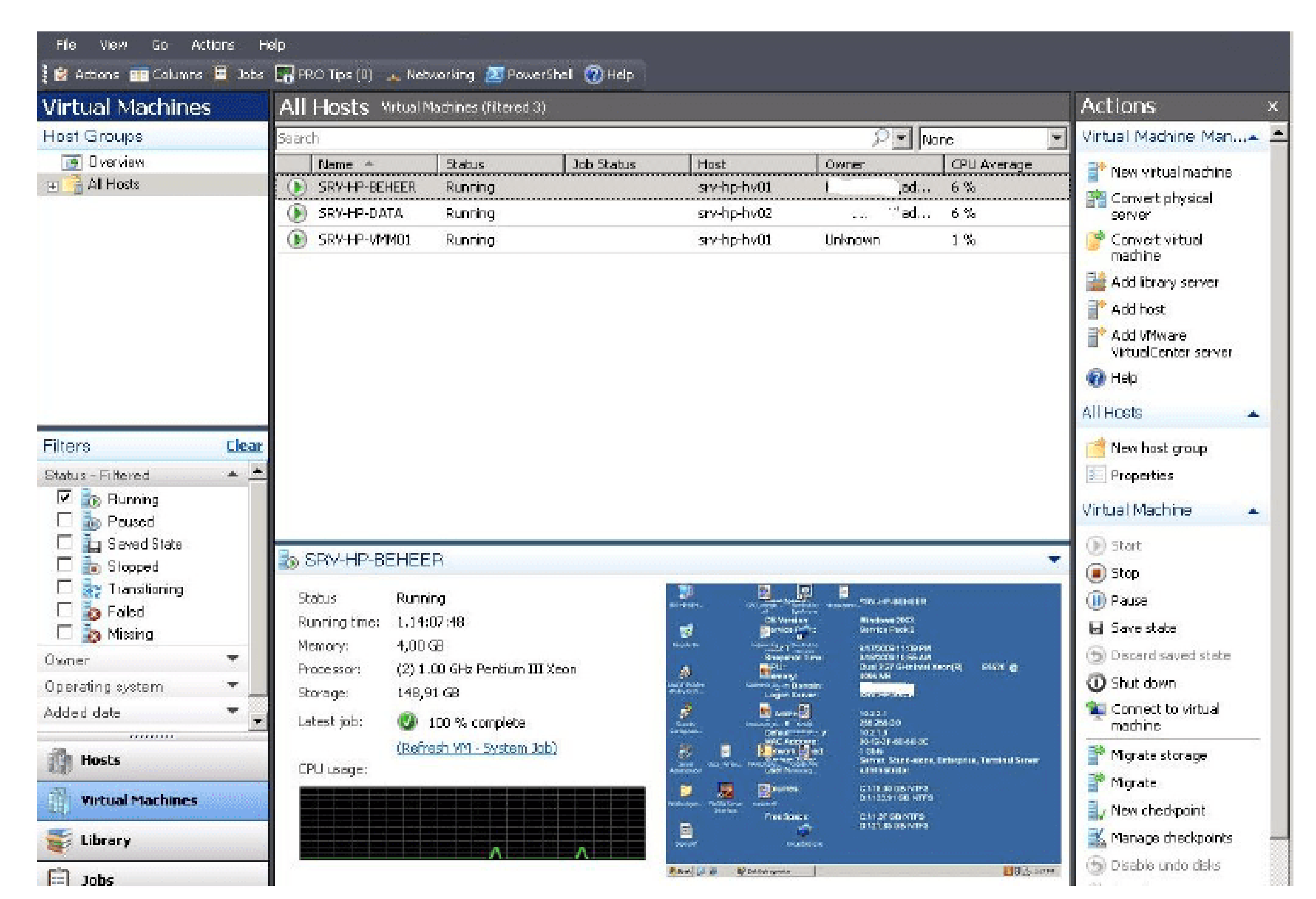
微软在本世纪初就察觉到虚拟机的重要性，因此也收购了当时唯一能和 VMware 抗衡的 Virtual PC。在工作站级虚拟机逐渐成熟，再加上竞争对手 VMware 在企业领域屡有佳作，微软体会到虚拟机无可避免将走入企业，因此在 2005 年即开始计划原生架构的产品。



▲ Hyper-V R2 是 Windows 2008 R2 中的一个角色

1. 从操作系统捆绑产品

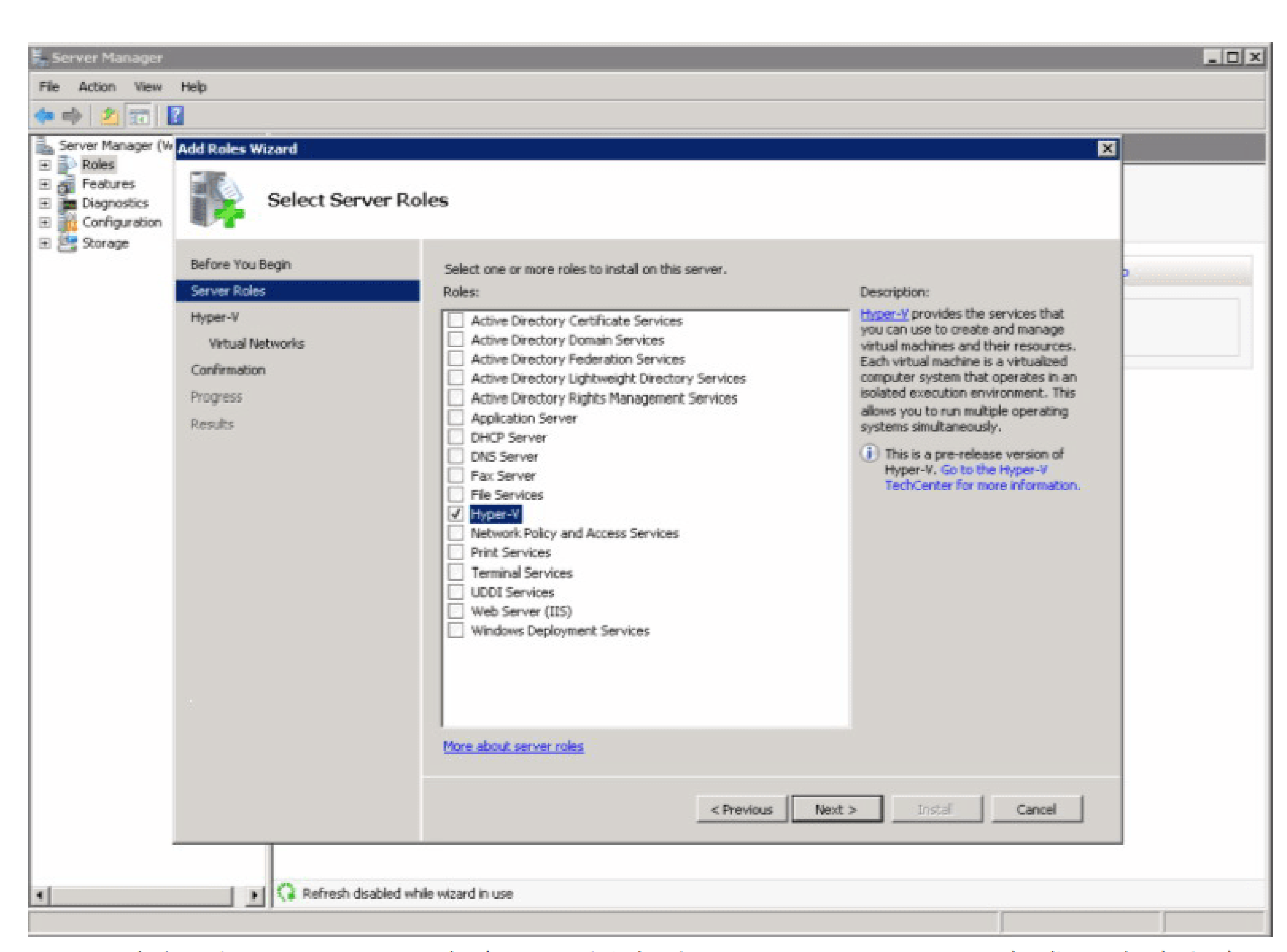
微软的策略很简单，用最大占有率的 Windows 操作系统的优势绑架产品逼迫用户就范。从 Office 打败 Lotus、从 IE 打败 Netscape 以来，这个手段就没有失败过，因此在企业虚拟机市场上，他们的策略也完全一样：使用主流操作系统来绑架虚拟机产品，当然这个产品也不能太差，而在企业虚拟机的部分，就是由 Hyper-V 来担此大任。



▲ 在 Windows 2008 R2 中，Hyper-V 可以从远程管理

2. Windows Server 2008 R2 中的角色

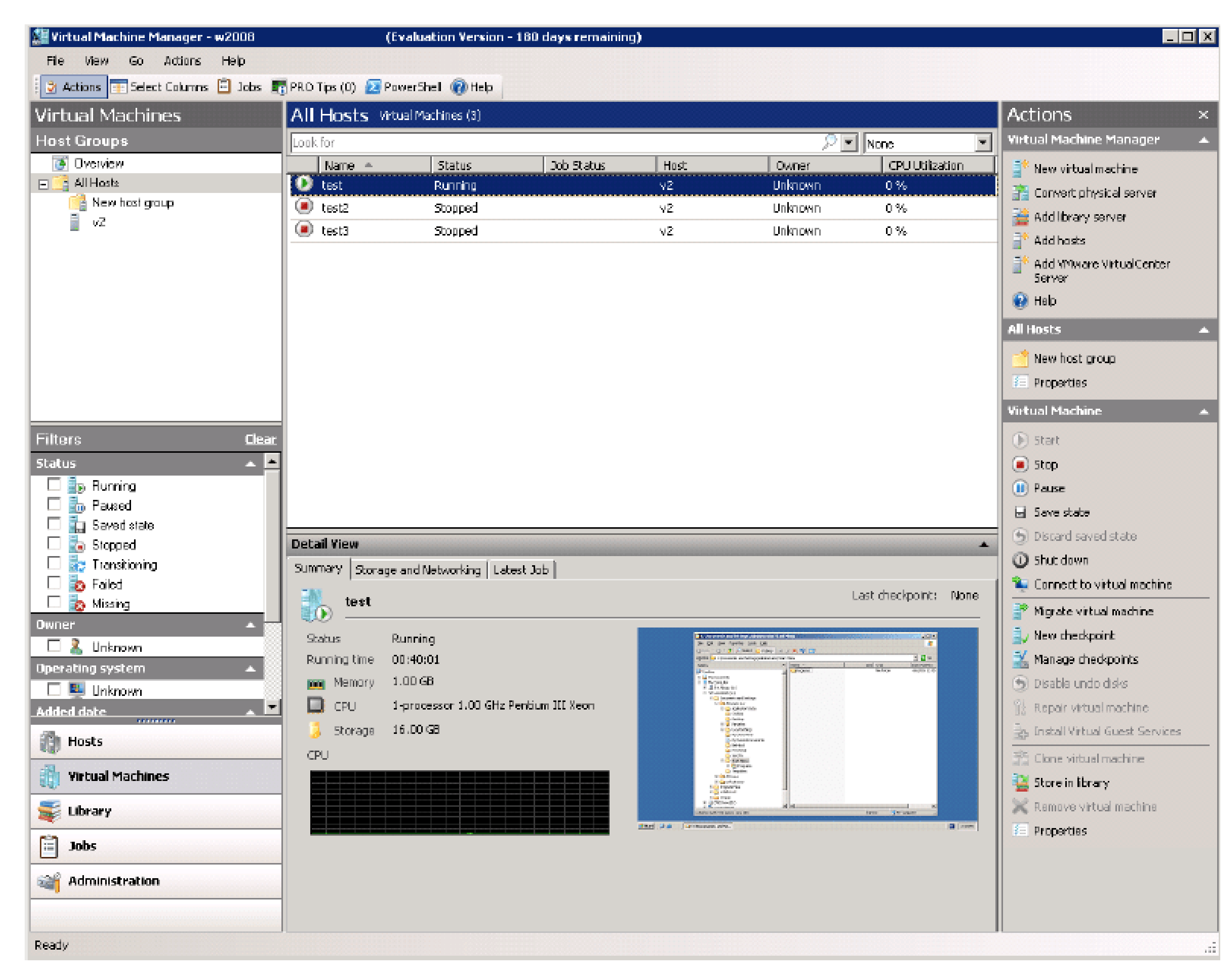
Hyper-V 是 Windows Server 2008 R2 中的一个角色，在将 Windows Server 2008 R2 提升成 Hyper-V R2 之后，引导后的 Windows Server 2008 R2 就不再是一个独立的操作系统，而是在 Hyper-V R2 上的一个客户端操作系统，但资源的分配还是可以由该操作系统来统一。



▲ 升级到 Hyper-V R2 角色后，原来的 Windows 2008 R2 变成一个虚拟机

Hyper-V R2 主要的功能和任何一个虚拟机产品一样，希望能将微软的服务器服务单个化，并

且充分利用物理机的资源。随着 VMware 的 vSphere 上市，微软虽然当前没有真正能抗衡的产品，但伴随着 Hyper-V 和早期的 Virtual Server，也推出了集成 Service Console 的 Virtual Machine Manager（SCVMM）。当前不但能管理微软的虚拟机，更可以管理 VMware 或是 ESX 下的虚拟机。



▲ 也可以通过 SCVMM 来管理

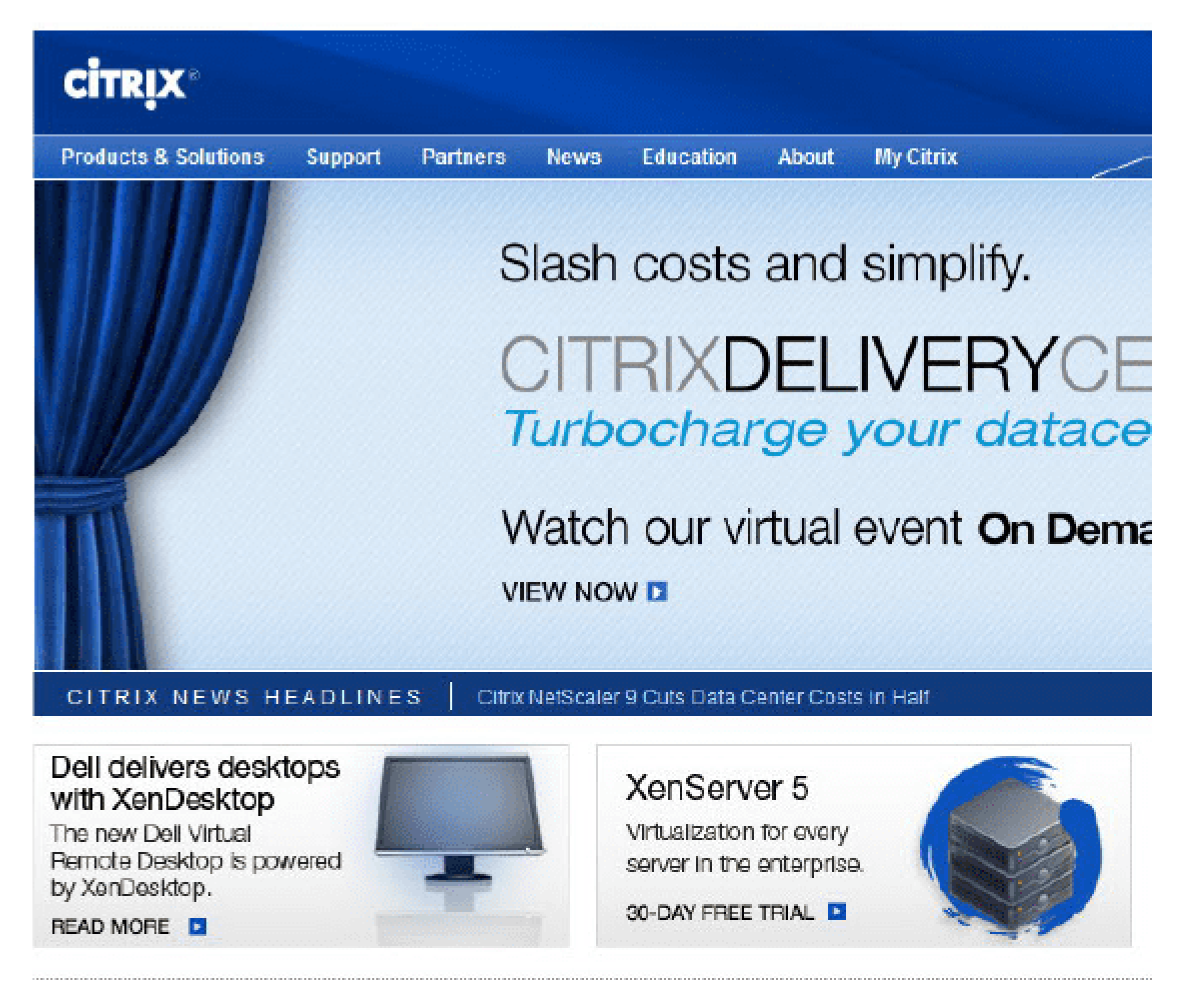
Hyper-V R2 的成败将决定微软在虚拟机领域的下一步动作，虽然当前还无法和 VMware 抗衡，但微软进入企业虚拟机市场的决心，让企业虚拟机的市场绝对无法平静。



▲ 微软的云计算平台 Azure 就是以 Hyper-V R2 作为虚拟机

1.5.3 最专业的桌面应用：Citrix Xen

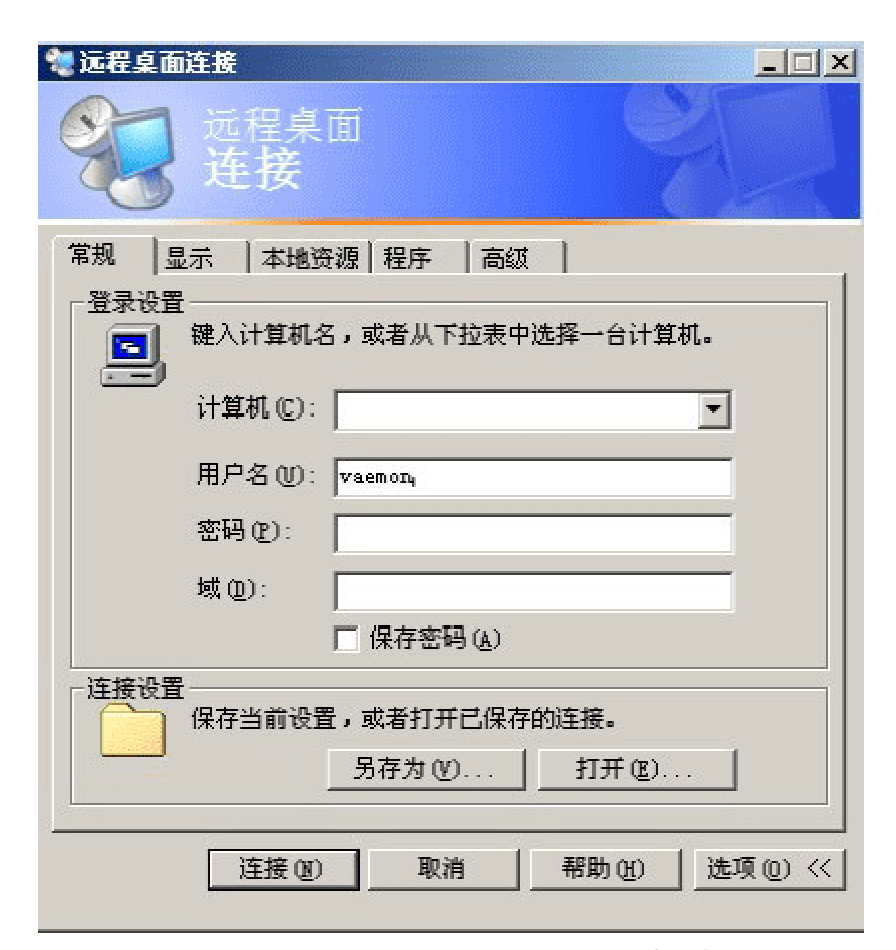
虚拟化的范围很广，许多厂家已投入多年。在桌面虚拟架构的领域中，最有名的就是 Citrix。企业统一桌面环境一直是大问题。不同职务的人需要使用一致的环境，如开发、财务、客服等。微软在 Windows 2000 之后就有终端服务，而 Citrix 就是将终端服务发挥到极致的厂家。



▲ Citrix 是以远程桌面起家的厂家

1. 从终端服务出发

然而终端服务有太多问题。首先就是操作系统的兼容性。终端服务大部分运作在服务器等级的操作系统上，但用户却习惯于桌面等级的操作系统。此外在安全性上，终端服务服务器的蓝屏，将让所有的用户都无法登录。在终端服务的使用已经贴近成熟时，这些问题的解决貌似仍遥遥无期。

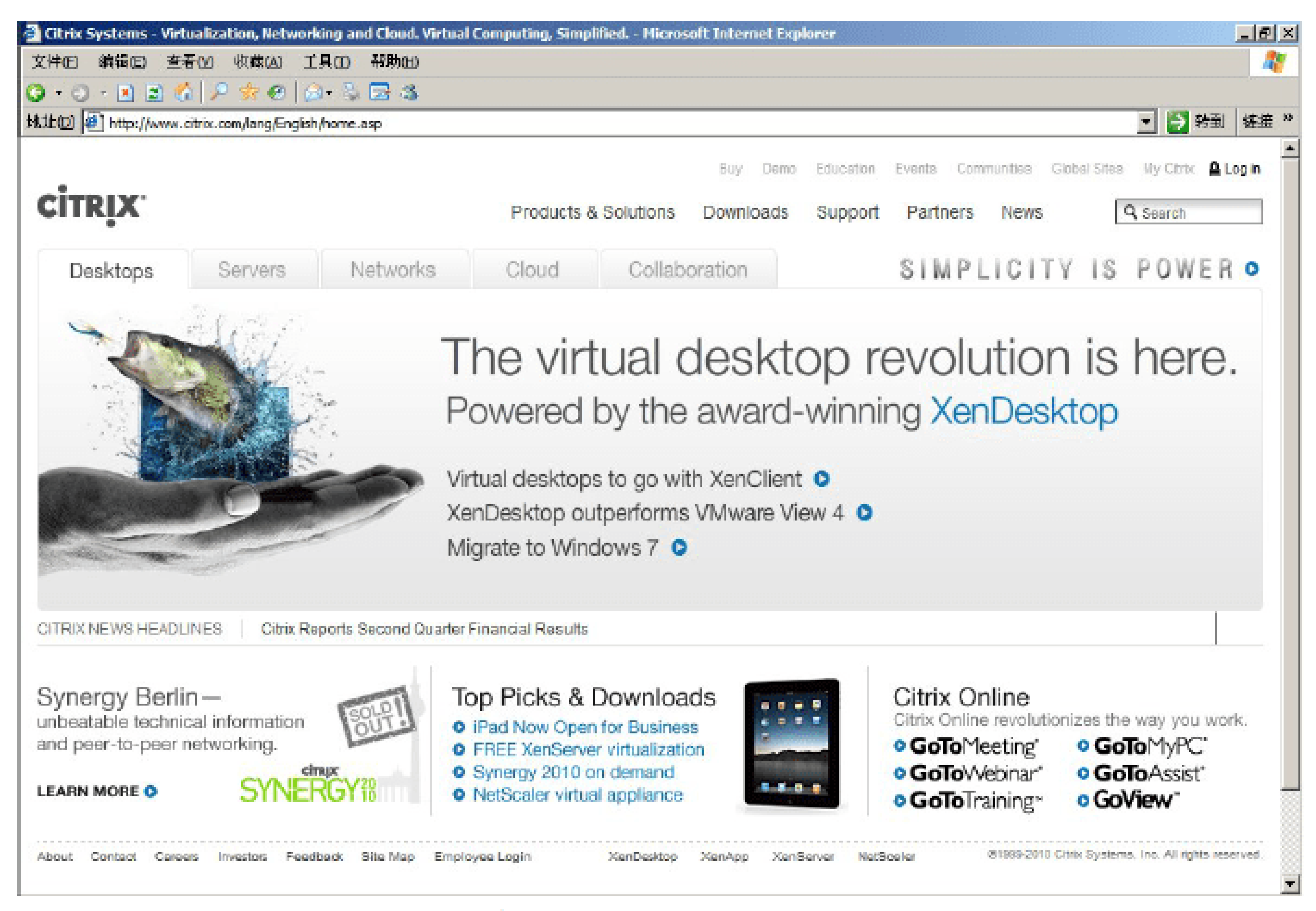


▲ Windows 在 2000 后就有终端服务了

2. 虚拟桌面架构的导入

虚拟桌面架构看来是这些问题的最好解答。在一个充满虚拟机范本的资源池中，让每一个终端用户连入单独的虚拟机而非终端服务服务器，这些虚拟机可能是 Windows 7 或是 Ubuntu Linux Desktop。同职务的用户可以使用一个模板，并且利用 Group Policy 和 Roaming 配置文件来规范本地及资源的使用，这种模式对 Citrix 这种以终端服务为主的厂家深具吸引力。

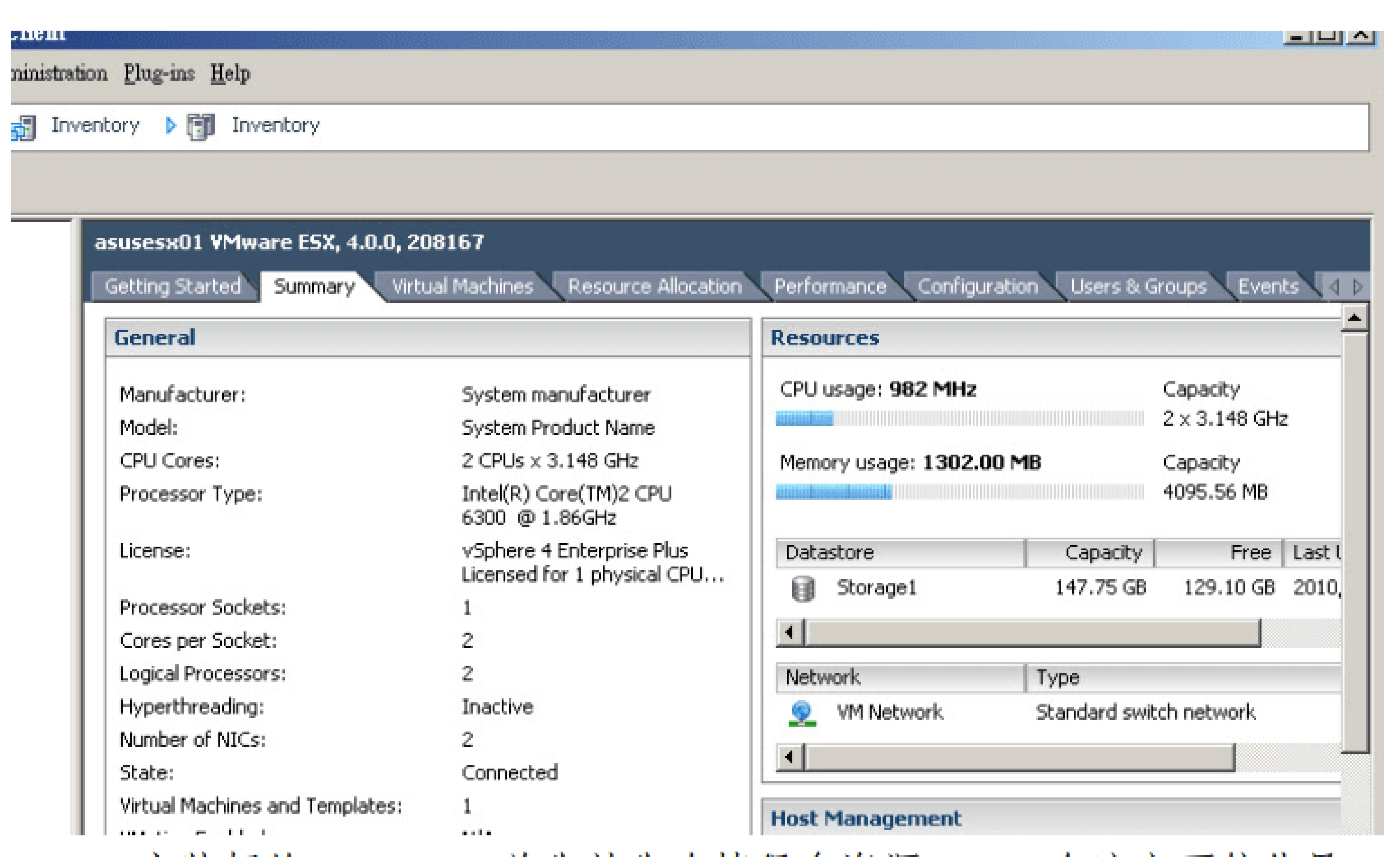
虚拟桌面架构模式的基础仍然是虚拟机，要实现 VDI 取代终端服务，要具备一个强而有力的虚拟机产品，因此 Citrix 也在 2007 年 8 月 15 日以 5 亿美元收购了 Linux 上的虚拟机龙头 Xen，作为 Citrix 健壮其桌面、解决专案领域以及进攻企业架构的强力武器。



▲ Citrix 的虚拟化市场是以 XenServer 为主

3. 最少的系统资源占用

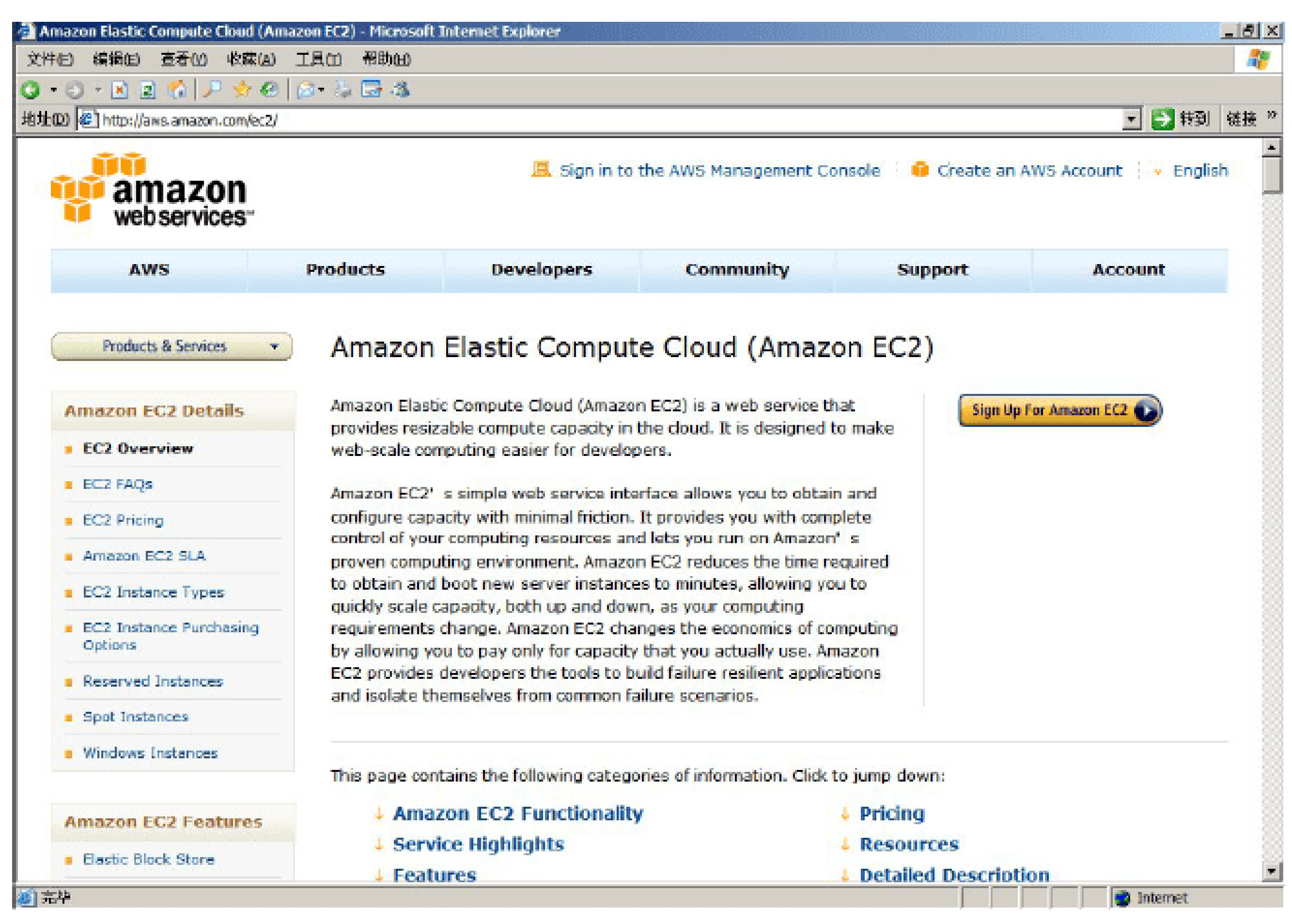
Xen 最有名的就是资源占用是所有主流产品中最小的。在一般情况下，Xen 的 Hypervisor 占用的系统资源在 2%，最大也只到 8%，这和大部分的其他虚拟机产品动辄到 20%的占用率比当然是好上很多。在 Intel 和 AMD 相继推出具有硬件命令及提速的 CPU 之后，Xen 也从善如流，不但充分利用硬件上的优势，新版的 Xen 也可以不修改客户端的操作系统直接使用，并发仍维持 92%～98%的资源利用率，让我们在 VMware 和微软之外又多了一个选择。



▲ 安装好的 vSphere，首先就先吃掉很多资源，Xen 在这方面较优异

1.6 虚拟化和云计算

vSphere 是一款云端系统，这句话让很多刚接触虚拟机产品的人一头雾水，虚拟化又和云计算扯上什么关系了？云计算也是一个用到烂的名词，2009 年更被定为“云端元年”，但云计算到底有什么不同？我们在这一小节就来看看云计算和虚拟化的关系。



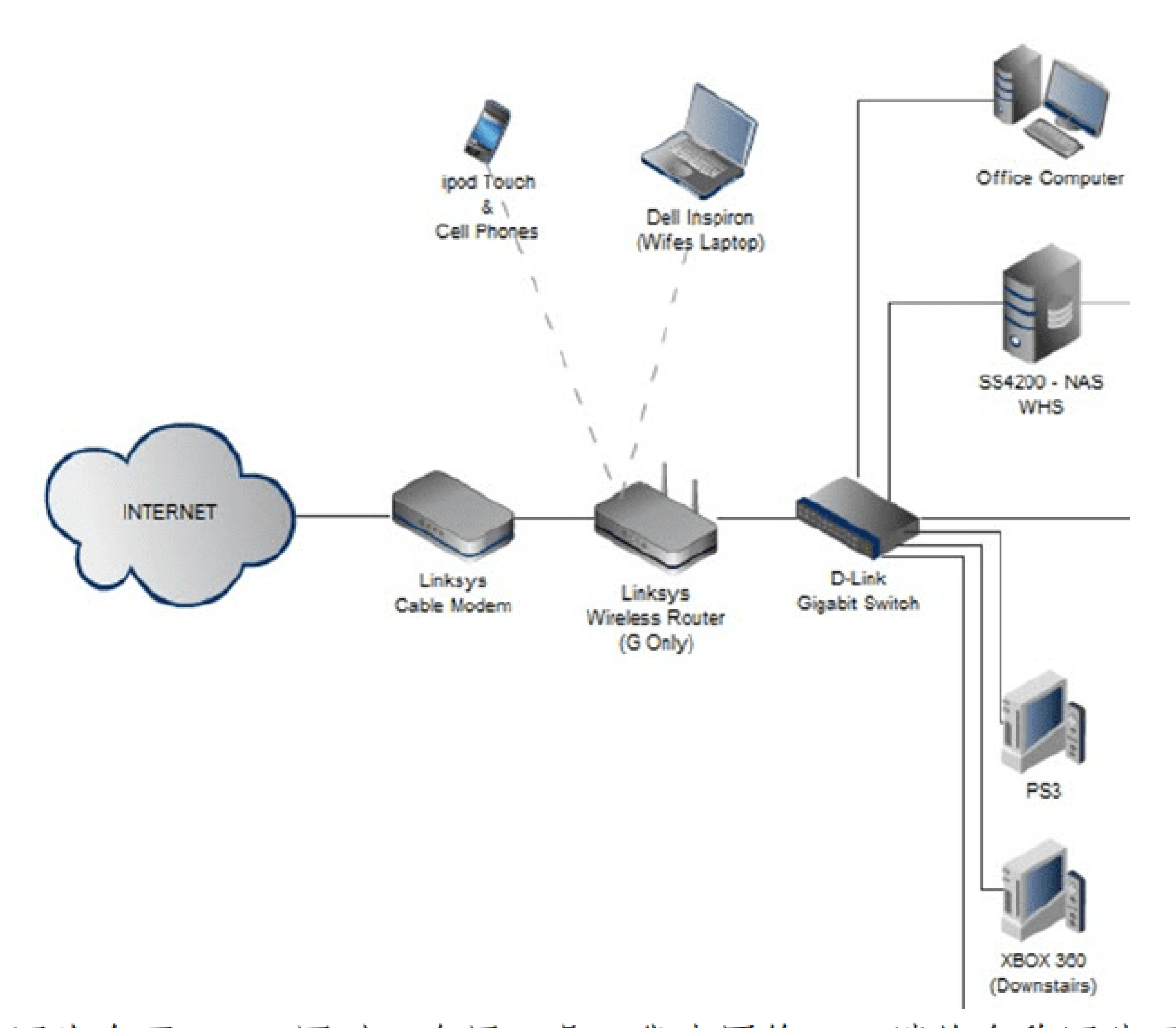
▲ 最常听到的云计算产品就是 Amazon EC2 了

1.6.1 什么是云计算?

云计算真的要说清楚，一整本书也说不完，但是仅以云计算和虚拟化之间的关系来说，可以用两句话来简单定义云计算，分别是：

- “将已量化的许多运算资源，以网络为提交媒介，通过特定的界面给需要服务的用户。”
- “将资源打散，找出最小的计量单位，之后将其全部集中起来，并重新分配的过程。”

笔者特别不用一些专有的名词来解释云计算，主要的原因就是云计算其实只是一个观念而不是一个技术或解决专案，只要能匹配上面两句话的应用，都可以称为云计算。



▲ 因为在画 Visio 图时，会用一朵云代表网络，云端的名称因此而来

1. 云计算的关键词

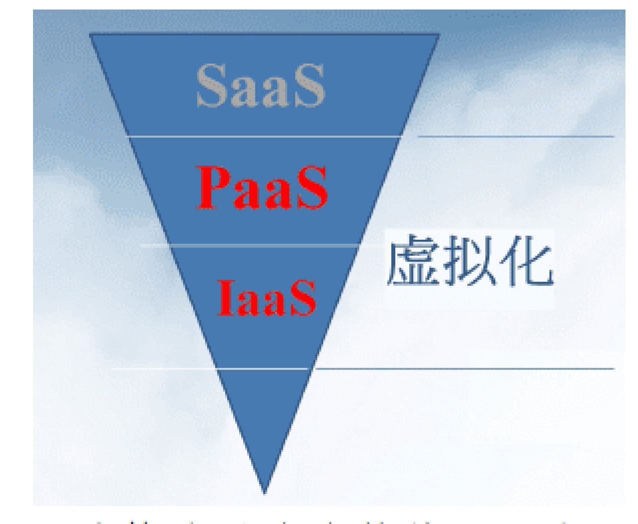
上面一句话中，我们可以看出其中的几个关键词，这些关键词就是创建云端运架构的基础，下面就是说明。

- 量化：方便计算，量化才能决定如何分配资源以及定义最小单位。

- 许多：认定单个资源不够，需要多个资源来落实副本及累加性能。
- 运算资源：在这边，我们将 CPU，RAM，网络联机及存储加壳成一个资源。
- 网络：有了网络才能到达每一个地方。
- 界面：有了界面才能提供通用性的服务。
- 服务：将资源加壳成服务的模样，方便计价及计量。
- 用户：需要服务的对象，不见得是人，有可能是机器或是其他的服务。

2. 云计算的层次

在之前曾提到计算机的架构层次，云计算也是计算机的一种，当然也匹配该层次架构。但云计算的层次较简单，一般分为软件（Software）、平台（Platform）和架构（Infrastructure）三个层次，而这三个层次的云端化，也分别有不同的产品及服务对应，我们称为软件即服务（SaaS，Software As A Service）、平台即服务（PaaS）和架构即服务（IaaS），而虚拟化的热点就是在 PaaS 及 IaaS 上。



▲ 云计算的层次为软件、平台及架构

1.6.2 不同层次云计算的说明

不同层次的应用有不同层次的手段，但基本上云计算不会离开前面的几个关键词，我们就在这一小节来看看几个有名的云计算。

1. 软件即服务

软件服务最有名的就是 Flickr。这个以照片分享为主的服务，在付费的部分，再显示其使用云端的痕迹。



▲ Flickr 是最标准的云计算的服务实例

▶ Flickr 的云计算

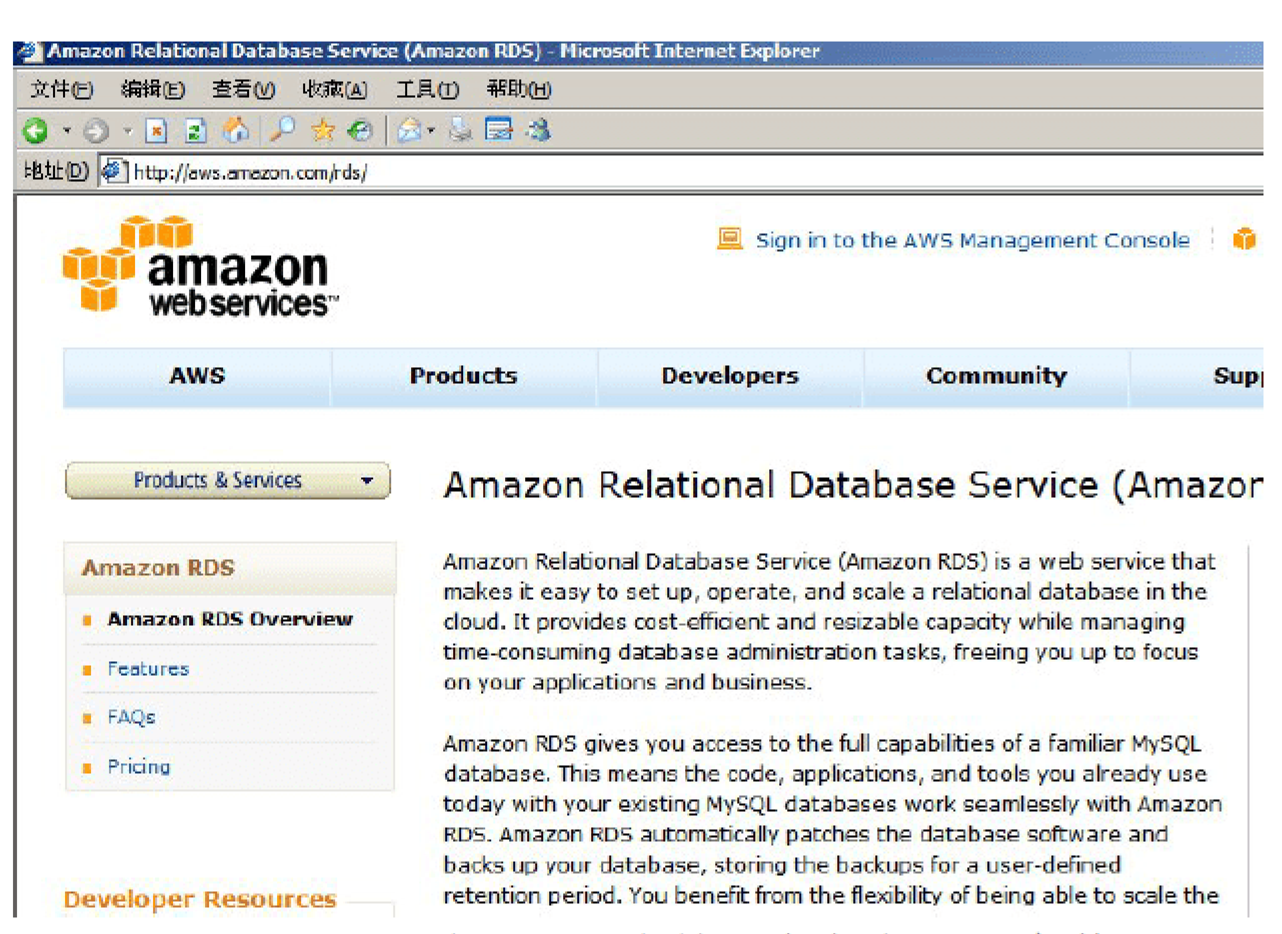
- 量化：以照片的数量（磁盘空间）及每个月传输的流量（网络带宽）作为其“量化”的基础。
- 许多：照片放在不同的服务器上（如 farm3、Farm5 等服务器），匹配了“许多”的条件。

- 资源：Flickr 所处的服务是一个 Web Service，当然有 CPU/RAM/网络/存储等资源。
- 网络：Flickr 是通过网络的服务，包括直接浏览和代码连接。
- 界面：Flickr 使用的是浏览器，以浏览器为其界面。
- 用户：Flickr 可以通过浏览器让人类看照片，也可以通过 html 代码来让网页将图片嵌入，服务的对象包括人类及网页。

2. 平台即服务部分

在 Platform As A Service 部分，最好的例子就是 Amazon AWS 中的 SimpleDB 或 Google 的 BigTable。首先定义 Platform 为整体的服务，如数据库服务、Web 服务等。而 Amazon 的 SimpleDB 就是标准的 Database Platform As A Service。

Amazon 的 SimpleDB 是一个网页界面的数据库，一般数据库的新建/修改/删除/查询都可以在网页上落实，也可以运行 SQL 语句。SimpleDB 是架构在一个称为亚马逊关系数据库服务（ARDS, Amazon Relational Database Service）的云端服务上，用户可以随时新建一个数据库，并且可以任意分配数据库的资源（如 CPU、内存等）。算是最标准的平台即服务。



▲ Amazon 的 ARDS 也是很有名的 PaaS 架构

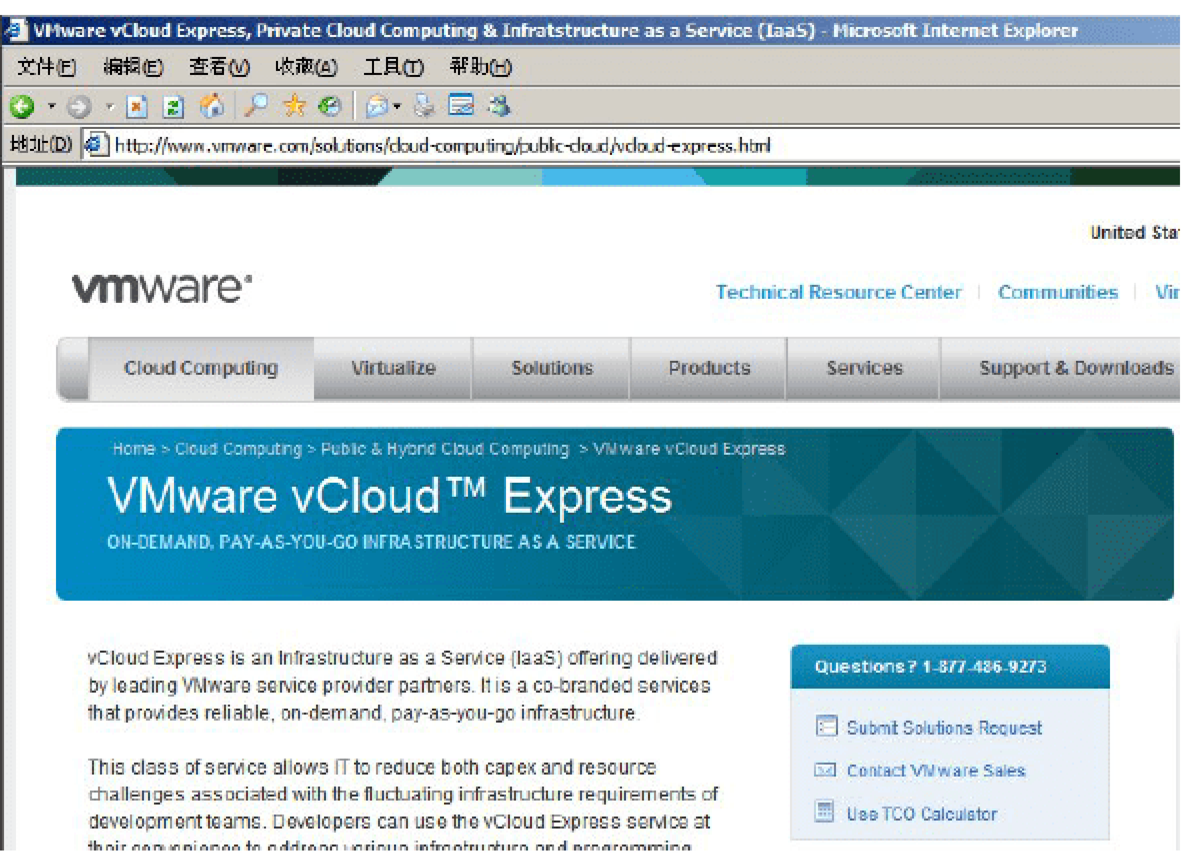
▶ 亚马逊关系数据库服务的云计算

- 量化：数据库的大小以及使用的 CPU/RAM 资源。
- 许多：数据库置放的主机可以做成 Cluster，支持多主机。
- 资源：所处的服务是一个 Web Service，当然有 CPU/RAM/网络/存储等资源。
- 网络：通过网络的服务，包括直接浏览和代码连接。
- 界面：操作界面使用的是浏览器，以浏览器为其界面。
- 用户：可以通过浏览器让人类操作数据库，也可以通过 SQL 代码让其他的服务访问数据库。

3. 架构即服务

2006 年时，Amazon 在信息业投下了一颗炸弹，就是有名的 EC2 服务。EC2 将 CPU、RAM、存储空间以及网络提交速度量化加壳成商品，让用户能通过网页在网卡创建自身的主机，并且可以选择操作系统，在这个操作系统中安装自身的服务。举例来说，你可以通过网页创建三台主机，分别是 Linux、Windows 2003 及 Windows 2008，并且安装自身的服务，通过远程桌面来操作，就好像操作自身的计算机一样，而这些计算机可以随时生成，随时消灭，不限数量，不限性能，可以给定多个 CPU，大量的 RAM 和可伸缩的存储空间，也可以决定网卡的数量和速度，这就是标

准的将架构视为服务的一部分，本书所提的虚拟化就是这部分。



▲ 除了 EC2 之外，vCloud Express 是以 VMware vSphere 为主的 IaaS

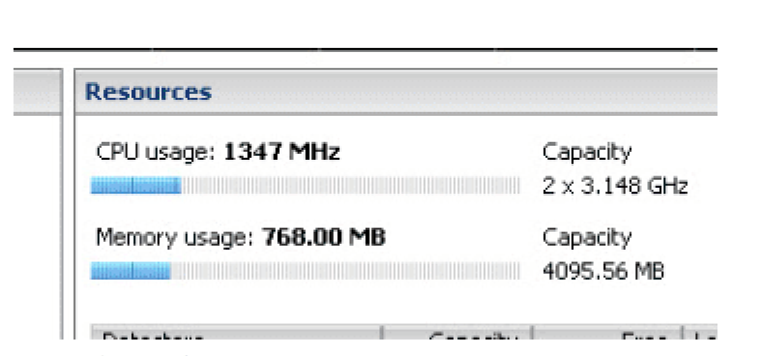
小常识	Amazon 的 EC2 的主机都是虚拟机，而这些虚拟机都是使用 Xen 所生成的。

1.6.3 虚拟化和云计算的完美结合

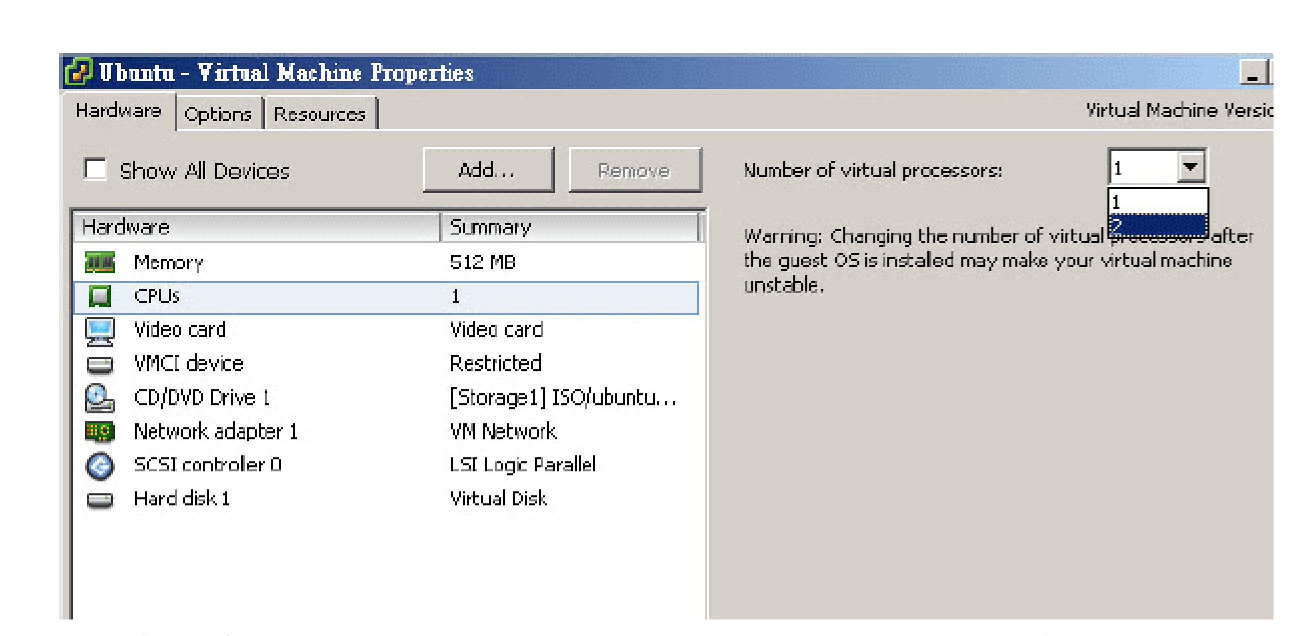
谈了这么云计算，云计算和虚拟机之间到底有什么联系呢？从 SaaS、PaaS 和 IaaS 中，我们可以看出主机的重要性。由于云计算对主机稳定性不信任，因此多台主机是必要的，但没有虚拟机也可以有多台主机，为什么一定要用虚拟机呢？事实上，虚拟机可以说是云端运维的基础，没有虚拟机就没有云计算，下面就来看看原因。

1. 虚拟机可定义最小的量化单位

单个物理服务器的资源太多会造成浪费。利用虚拟机定义出最小的量化单位，当资源不够时，以虚拟机为单位来添加运算能力（使用群集），当资源超过时可以减少主机数量，这些功能没有虚拟机是做不到的，因此我们利用虚拟机来创建主机基本运算单元，再将这些单元以积木的方式来排列组合，可以让资源获取最大的应用。



▲ 虚拟机中以 MHz 来分配 CPU 的速度

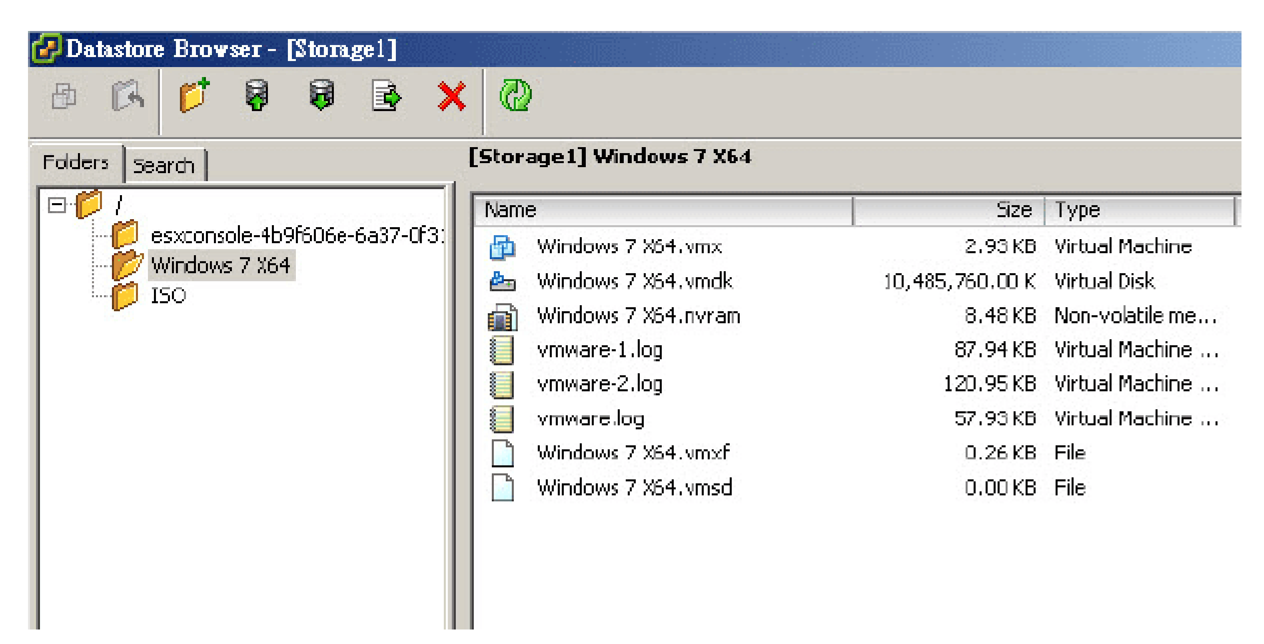


▲ 虚拟机中也以 vCPU 的数量来分配，可真正做到资源划分

2. 快捷生成或消灭资源

云计算最重要的特色就是使用浏览器，但要利用浏览器来生成或消灭物理机是无法办到的。虚拟机是创建在软件的 Hypervisor 上，只要是软件，就可以利用 API 来回馈到浏览器，因此我们

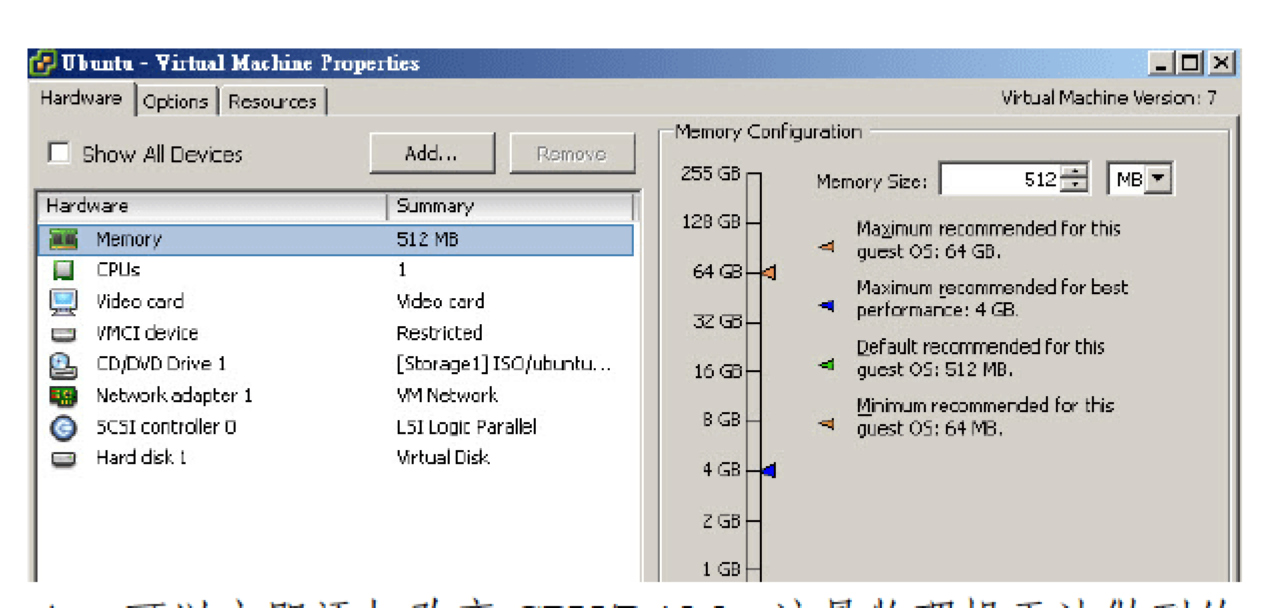
可以在浏览器上添加一台主机，当资源太多而引起浪费时，只要将虚拟机代表的文件删除或关机即可。这对云计算的 PaaS 或 IaaS 都是最重要的功能，只有虚拟机能办到。



▲ 在虚拟机中，一台 VM 就是一个文件而已

3. 在线配置软硬件

有时我们的主机资源不够，但还没有不足到要多台主机时，这时加 CPU 或是 RAM 可以解决问题。但服务又不能停止怎么办？虽然当前有 Hot-Add 的主板，但价格昂贵又不一定能在线配置；但使用虚拟机的话，可以在线直接配置 CPU/RAM/存储空间等，更可以在引导状态时进行这些操作，这些都是物理机在云计算中无法胜任的主要原因。



▲ 可以立即添加改变 CPU/RAM，这是物理机无法做到的

1.7 信息从业人员的新挑战

最近和几个网管人员聊天，惊讶的是发现他们心中的虚拟化竟然就是单纯的 VMware Workstation，甚至他们觉得 VMware 不是一个公司而是一个产品，这一点让笔者非常惊讶。在软硬件日趋复杂的今天，信息人员面临的挑战将会是大量的服务器、无数的虚拟机以及复杂的企业网络架构，信息人员在云端时代的今天，一定要拿掉“网管”的观念，来迎接理解虚拟化和云计算的新挑战。下面我们就从不同的角度来看看。

1.7.1 平台相关技术

在虚拟化流行之后，包括了 PM、CIO 或是网管人员，要学习的技术比从前更多了。笔者在此先强调，在前几章遇到不懂的名词或是观念没关系，本书的稍后都会有详细的说明。

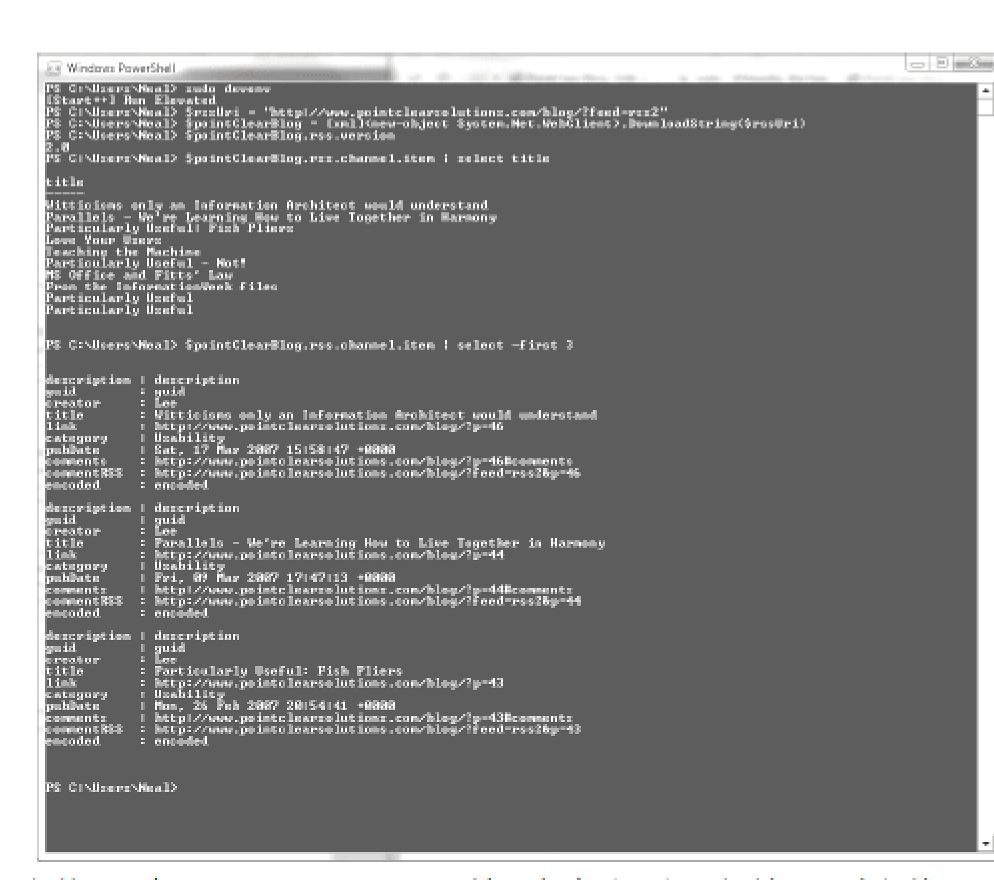
1. 网管以外的知识

除了网管的基本技术之外，从前不需要学习的各种平台和架构的使用都需要掌握了，因为在

学习了虚拟化之后，常常会有一些无法预期的问题，这就首先必须了解是 VM 还是平台本身问题，因此多理解平台才能知道错误发生的原因。

```
vacmonco@vacmon:~$
vaemonco@vaemon:~$ ls -al
total 93804
drwxr-xr-x 4 vaemonco vaemonco     4096 2010-03-18 15:04 .
drwxr-xr-x 4 root     root         4096 2009-04-05 18:07 ..
-rw------- 1 vaemonco vaemonco     7560 2010-03-11 14:52 .bash_history
-rw-r--r-- 1 vaemonco vaemonco      220 2009-04-05 18:07 .bash_logout
-rw-r--r-- 1 vaemonco vaemonco     3115 2009-04-05 18:07 .bashrc
-rw------- 1 vaemonco vaemonco        9 2009-06-22 21:35 .nano_history
-rw-r--r-- 1 vaemonco vaemonco      675 2009-04-05 18:07 .profile
lrwxrwxrwx 1 vaemonco vaemonco        8 2009-04-05 21:45 public_html -> /var/www
drwx------ 2 vaemonco vaemonco     4096 2009-08-08 17:13 .ssh
-rw-r--r-- 1 vaemonco vaemonco        0 2009-04-05 18:12 .sudo_as_admin_successf
ul
-rwxr--r-- 1 vaemonco vaemonco      836 2009-08-10 23:49 syncblog
-r--r--r-- 1 vaemonco vaemonco 95909121 2009-11-17 11:15 VMwareTools-8.1.3-20373
9.tar.gz
drwxr-xr-x 7 vaemonco vaemonco     4096 2009-10-22 15:30 vmware-tools-distrib
vaemonco@vaemon:~$ _
```

▲ Linux 命令的熟悉是必要的技能



▲ 如果能理解 PowerShell，将对虚拟化的管理/操作更有帮助

▶ **平台的技术**

- Linux 使用。
- Windows Server 的管理。
- Active Directory 的管理及维护。
- 各种平台的使用及管理，如数据库，Exchange，IIS，Apache，TomCat 等。

2. 虚拟机软硬件的基本知识

要理解的虚拟化产品包括软硬件，我们先来看必须学习的软件部分。



中文比较vSphere和Hyper-V R2 I…

这边是指插槽不是核数，如4WAY的伺服器就是4个插槽，4个CPU。Hyper-V是跟Windows 2008的，因此Windows 2008是多少，Hyper-V就是多少。重点是，目前"垂直升级(scale up)"已经不是潮流了，用多台主机就能达到更棒的效能和管理功能，成本又较低，因此一台主机上插64个CPU的意义不大，因此这一点谁胜谁负没有太大意义。

最大主机CPU核心数：VMware胜(64 对N/A)

除了插槽之外，能被作业系统认出的CPU才是真正能用的CPU。目前多核心系统横行，因此在一个作业系统下能被认出并且分配资源的CPU数当然和核心数有关。VMware最大可以认出64个pCPU，Hyper-V应该也是Windows 2008的限制。

每CPU最大核心：无胜负(12 对 N/A)

Scott没说太明白，但嘉评认为这是硬体厂商的限制，作业系统只需拿到硬体CPU的设计就可以开发出相对应的驱动程式或作业系统核心，一个CPU有多个核心将会是以后的潮流。

最大虚拟机CPU：VMware胜(8 对 4)

vCPU是新版虚拟机产品必备的功能。有些应用程式可以利用到多CPU，因此在虚拟机中，也可以分配多个CPU资源给虚拟机中的应用。VMware能到达8个虚拟CPU(vCPU)，但老实说，一个虚拟机中有4个vCPU已经很够用了。

最大主机RAM：Hyper-V胜(2048 对 1024)

一台主机上能安装多少RAM，得看晶片组以及CPU的设计。以现阶段的应用来说，这些数字超过128GB就没有太大意义了，因为以目前CPU的运算能力来说，就算是4X4核的主机，上面能运行的虚拟机应该也不会超过200个(也没有人敢这么搞)，那么如果一台VM有512MB的话，也不过是100GB，因此能安装上2048GB又如何？当然有这个功能弹性较大，因此还是一个优点。

虚拟机最大RAM：VMware胜(255 对 64)

▲ 不同虚拟机产品的功能对比要多留意

▶ **虚拟化软件技术**

- 产品之间的差异，如支持的客户端 VM 操作系统等。
- 虚拟化产品的规格，如支持的 CPU 数、内存大小等。
- 提供的功能，如 VMware vSphere 提供 VMotion、Hyper-V 提供 Live Migration。
- 虚拟化产品的安装、校调、配置、升级、维护。
- 虚拟机的管理。
- 虚拟机上各项性能指针的跟踪。

- 硬件的兼容性及性能，如哪些芯片组或 CPU 支持 Intel VT-d。
- 硬件的配置，如一台物理服务器需要配置多少 RAM 才能运行 N 多虚拟机。

3. 存储设备及实践

存储是虚拟化之母，没有存储设备的虚拟化顶多只是 OS in OS 而已。VMware 的母公司 EMC 就是一个纯存储设备厂家。当前存储设备除了理解产品、规格等差异外，还要有规划及操作、副本、维护的能力等。我们会在本书稍后花很大的篇幅以实例说明存储设备在虚拟化中的应用。



▲ 理解不同的存储设备，现在甚至个人用的 NAS 都可以应用在虚拟机上

▶ **理解存储设备及实践**

- 什么是 NAS，NFS，DAS，FC SAN？
- 什么是 SAS，SATA，SCSI？
- 什么是 Logical Volume，什么是 LVM，如何设计？
- 什么是 LUN，什么是 RDM，HBA？
- 其他和虚拟机相关的知识。
- 什么是多路径（MultiPath）？

4. 网络设备及原理

企业在虚拟化之后少了操作和管理的复杂性，但相对的代价就是设计时的复杂。在虚拟化之后，整个企业的架构变得更复杂，而其中最麻烦的就是网络的管理。因此除了要理解原来的网络原理，操作交换机、防火墙之外，更要理解虚拟化的网络原理。本书稍后也会有完整的介绍。



▲ 配置交换机也是虚拟化的重要工作之一

▶ **理解网络设备及原理**

- 什么是 Switch，Router？
- 什么是 vSwtich，vRouter，vNIC？
- PortGroup 是什么，VM Kernel 是什么？
- 什么是 Trunking，Teaming？
- 设计及规划 vLan？
- 操作 Cisco Nexus 1000v 虚拟交换机？

结 语

本章的热点在于虚拟化的基本知识，也提出了当前和虚拟化最相关的云计算技术，从下一章开始，我们将介绍本书的主角 VMware vSphere，看看 vSphere 的强大功能，并且理解其如何融入当前的云端及企业环境。

读书笔记

第 2 章 虚拟化的首选：VMware vSphere

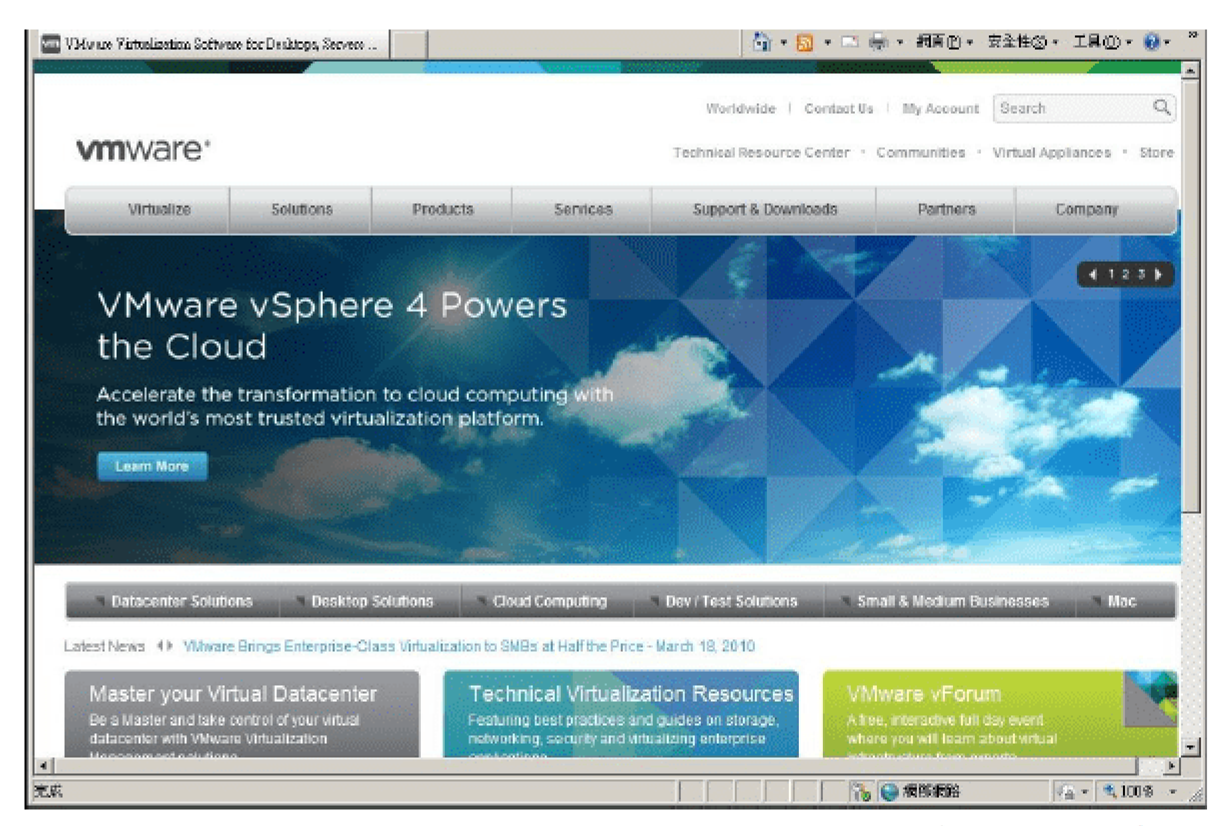
关键词：

- VMware 的历史
- VMware 的产品种类
- VMware vSphere 的架构
- VMware vSphere 的功能简介
- VMware vSphere 的版本分别
- VMware 的成本估算

在理解虚拟机的概况之后，本书就直接进入热点，看看主角 VMware vSphere 的全貌。虚拟机的观念简单，但实践起来并不单纯，还需要各种软硬件设备的配合，但就算软硬件设备都齐全了，也难保能设计出一个完善的 IT 基础架构。因此在真正动手之前，将整个基础架构、灵魂 VMware vSphere 做一个最全面的理解，才能打下良好的根基，让企业虚拟化立于不败之地。

2.1 理解 VMware 的虚拟化产品及公司

本书的主角是 VMware vSphere，VMware 这个名字已经很响亮了，但很多人对 VMware 还是有很多误会。VMware 是一个公司名称，就像微软也是公司一样，VMware 旗下有很多产品，就像微软旗下有 Windows、Office、Exchage、SQL Server 产品一样。但 VMware 的所有产品都和虚拟机有关，因此在虚拟化的产品在线，VMware 可说是全球霸主，也让 VMware 成为全球第三大的软件公司。首先我们就来看看 VMware 是如何一步步走过来的。



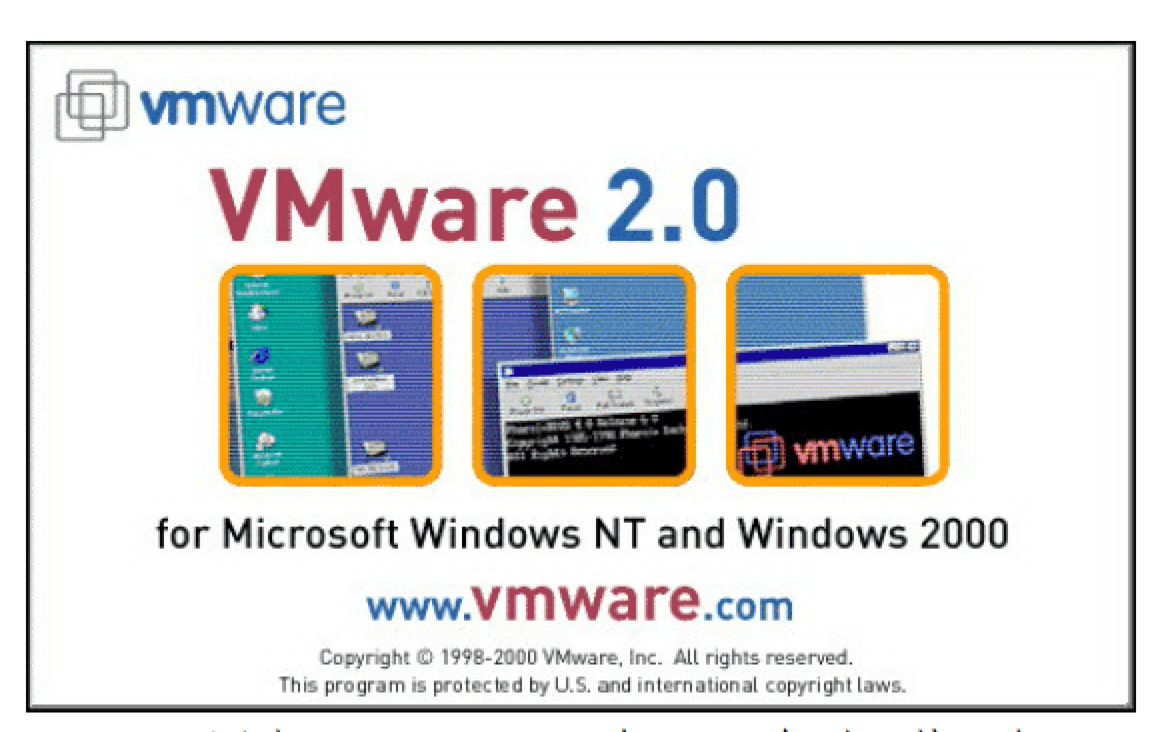
▲ VMware 的产品线十分完整，都是和虚拟化有关

2.1.1 VMware 虚拟机产品简史

虽然虚拟机产品在 1960 年就提出，但 X86 架构下的虚拟化一直到 VMware 出现之前都没有真正可用的商业产品。

1. 第一个虚拟机产品：VMware

在 1999 年时，VMware 推出了第一套 X86 架构的虚拟化产品。当时 VMware 就叫 VMware，但本质就是现在的 VMware Workstation，这也是许多人到当前为止一提 VMware 都以为就是 VmwareWorkstation 的原因。VMware 一上市就打响了名声，直接成为虚拟机的代名词。

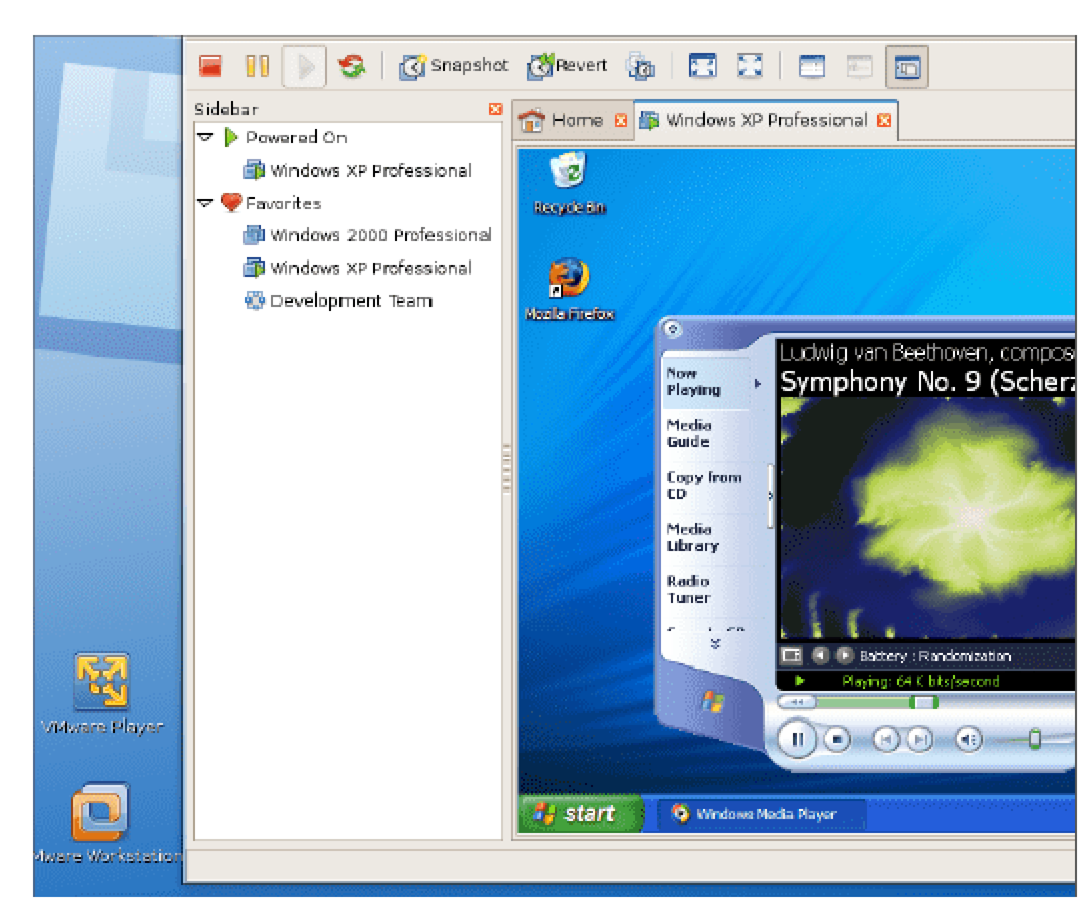


▲ 图为 VMware 2.0，在 2000 年时一炮而红

2. 从 Windows 平台走向 Linux

由于 Windows 系统在当时还是 Windows 98/2000，其 Memory Leak 和蓝底白字的蓝屏画面，让使用寄居架构的 VMware 充满了不确定性，这时 VMware 的工程师们，开始将目光放在了相对

较稳定的 Linux 身上，因此在 VMware 成功的不久之后，也出现了 Linux 版本的产品，让 Linux 下的用户也可以并发运行 Windows 或其他的操作系统，不过此时的 VMware 产品都是寄居架构的。

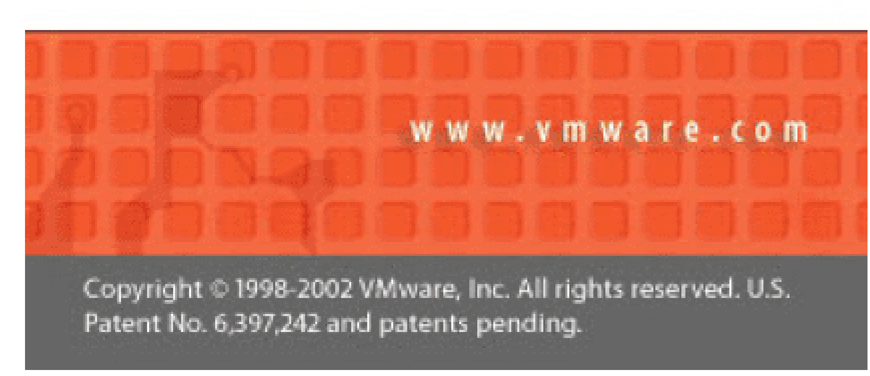


▲ 在 Linux 下运行的 VMware

3. 从寄居架构走向原生架构

在推出 Linux 版本的 VMware 之后，虽然稳定性大增，但 VMware 还是一个寄生架构的产品，无法摆脱寄居在固定 OS 上的恐惧。既然选择了较为稳定的 Linux，VMware 也顺势推出了运行在 RedHat、Mandrake 和 SUSE Linux 下的 VMware GSX Server，并且也拥有 Web 端的管理和客户端的管理程序。虽然功能不断加强，但没有一个真正自身能掌握的 Hypervisor，VMware 的发展仍然有相当大的制约，因此在 GSX 之后，VMware 打算开发一个真正自身的 Hypervisor，但同时也保留寄居架构的产品。与此同时，VMware 也将产品线的名称确定，原来统称 VMware 的产品，也正规命名为 VMware Workstation。





▲ VMware 的 Workstation 版本 3.2

4. 迎接原生架构的 Hypervisor

一劳永逸解决寄居在其他 OS 上的同时是 VMware 工程师的梦想。VMware 也在经过了多年的准备及开发之后，终于在 2005 年推出了自身的 Hypervisor，称为 VMware ESX Server。

ESX 以 Redhat 7.2 为基础，插入了自身的硬件内核，这个内核是由 Dr. Mendel Rosenblum 开发的 SimOS 演化而成，自此成为了一个真正原生架构的 Hypervisor（或称虚拟机 OS），而 ESX 的出现，正规声明了 VMware 踏入企业界的领域。



▲ VMware ESX 3.5

5. 全新的企业平台：Virtual Infrastructure

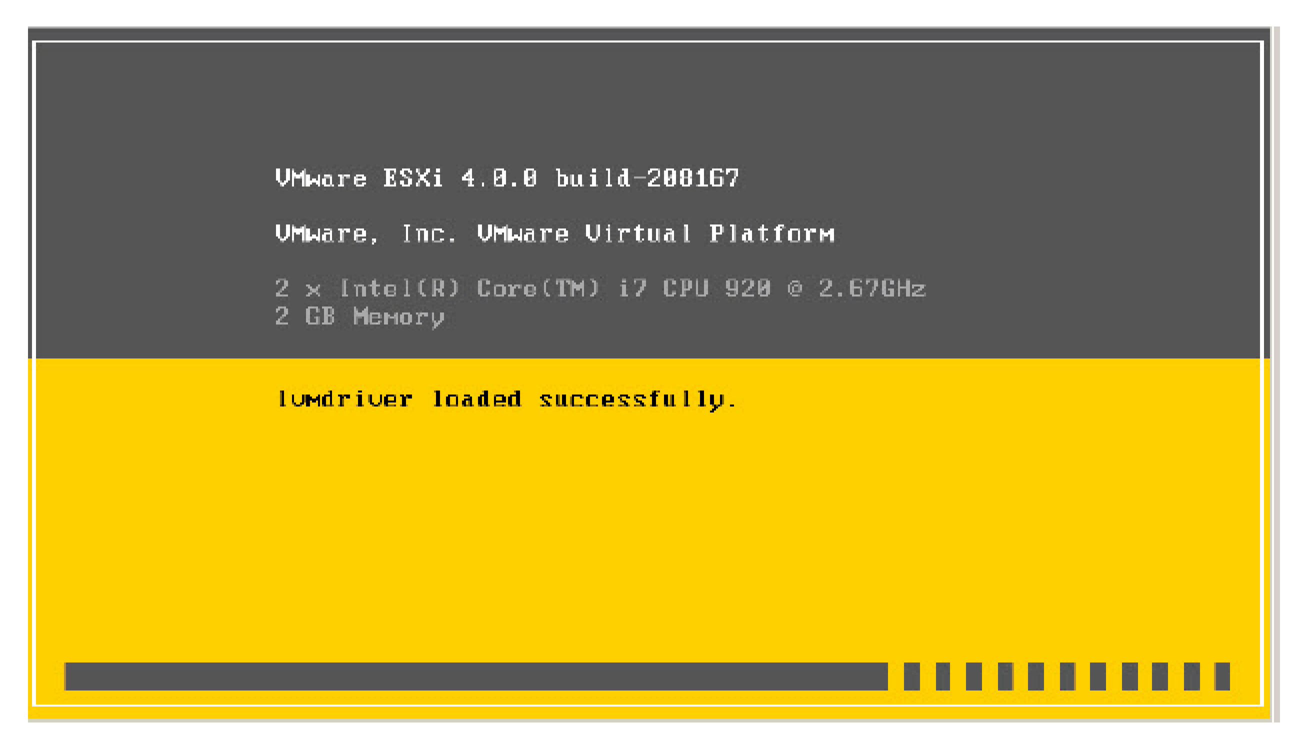
原生架构的虚拟机能做的事竟然超乎原来预期之外，而多台 ESX Server 在企业管理上也能

满足更多的业务需求，因此 VMware 在 2006 年时，围绕着 ESX Server 也推出了类似 ESX 集群的 Virtual Infrastructure 架构，简称 VI3。当然在集群出现之后，客户端的管理、集群之间的管理等基本功能也加入了 VI3，使得 VI3 成为全球唯一最完整的虚拟机集群产品。

6. 走向云端的 VMware vSphere

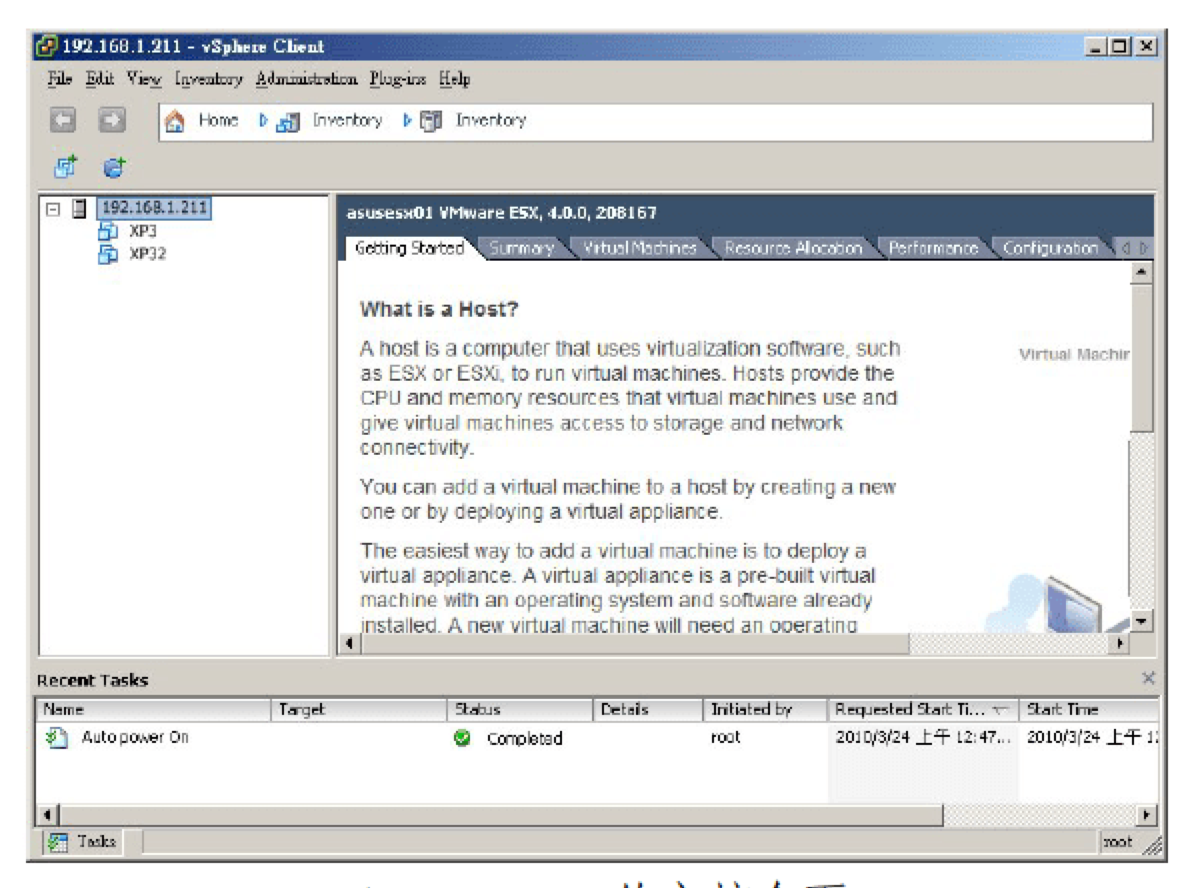
在 2000 年之后，全球的软件厂家都看准了虚拟化这一块市场大饼。微软收购了 Virtual PC，Sun 买下了 Virtual BOX，Citrix 买下了 Xen，这让 VMware 的压力越来越大。同一时间内，巨型网站也开始进入云端化；Amazon 在 2006 年推出 EC2，使用 Xen 的虚拟机，更让 VMware 感受到用户无法满足单个虚拟产品或架构。

2009 年 4 月，VMware 终于在等待了三年之后，推出了全新云端架构的 VMware vSphere，为了和之前的产品区别，VMware vSphere 也被称为 VMware Infrastructure 4.0（VI4）。



▲ ESXi 4.0 的引导画面

VMware vSphere 的基本还是原生架构的 ESX/ESXi，但整个系统的中心已不再是 Hypervisor 了。VMware vSphere 以云端系统为基础，以 Hypervisor 为工具，发展出各式各样的功能，这和 10 年前的 OS in OS 已完全不同，而 VMware vSphere 出现，一举让 VMware 成为当前市面上最完整的云端操作系统，本书所讲述的所有的功能都围绕着 VMware vSphere 展开。



▲ ESX 4.0 的主控介面

2.1.2 理解 VMware 这家公司

麻省理工学院（MIT）和加州大学柏克莱分校可以说是现代科技的摇篮，VMware 就是出自

MIT，而 VMware 的创办人也都毕业于柏克莱大学。



▲ Greene 女士和 Rosenblum 博士等人创立了 Vmware

1. 草创时期

早在 1998 年时，毕业于 MIT 的 Diane Greene 女士已然察觉计算机资源的使用率过底，因此和 Dr. Mendel Rosenblum、Scott Devine、Dr. Edward Wang 以及 Edouard Bugnion 等人成立了 VMware 公司，当时还是只专精于 OS in OS 的软件，以期待能发挥硬件的完整资源，并且也提供当时专业 IT 人员一个测试、评估的低成本环境。



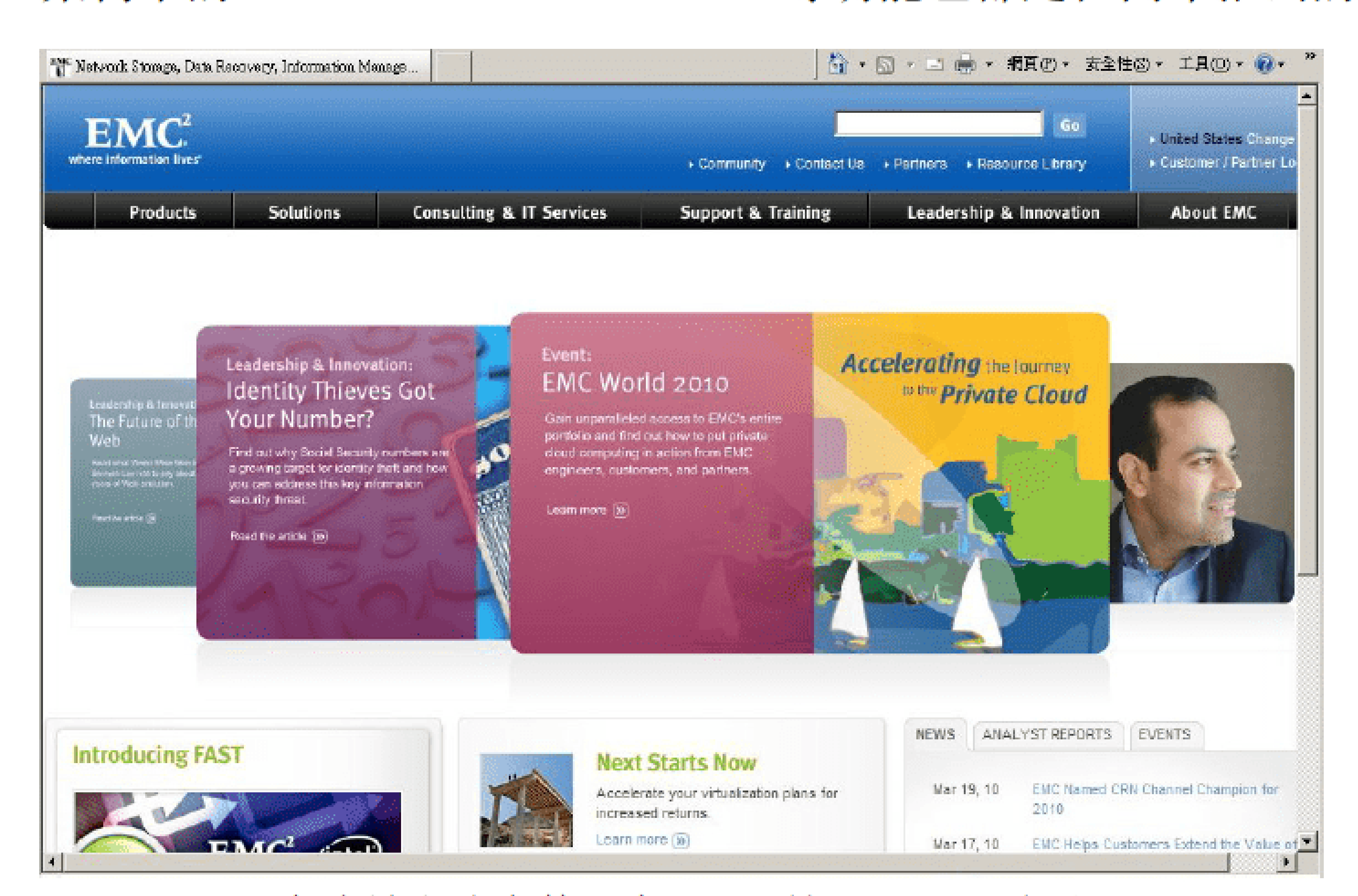
▲ 为 Palo Alto VMware 的公司原址

2. 兴盛时期

自 2000 年后，靠着 VMware Workstation 的名称，VMware 开始成长，公司的办公地一直在北加州的 Palo Alto，也是美国科技的重镇。也先后在 MIT 及纽约设有研发中心。在 2005—2006 年之间，VMware 也陆续推出了在 Windows、Linux 以及 Mac OS 下运行的虚拟化产品，更跻身于财星前 100 大企业。

3. EMC 收购

随着不停的发展，VMware 也避免不了被大公司收购的命运。2004 年，存储大厂 EMC 全额收购了 VMware，当年的 VMware 是全盛时期，不但推出了各项产品，更推出了 64 比特的支持，而后期出现在 VI 架构中的 vCenter、SMP、VMotion 等功能也都是在同年推出的。



▲ 全球最大的存储厂家 EMC 是 VMware 的母公司

4. 股票上市时期至今

2007 年 8 月，发展迅速的 VMware 在 EMC 释出 10%的股份，在纽约证交所正式上市，成为第一个上市的虚拟机产品公司。自此，VMware 站稳全球第三大软件公司位置，并不断推出叫好又叫座的产品，虽然面对环伺四周的强势，VMware 仍然是虚拟化产品的霸主，直至今日，VMware 仍然占有 65%以上的虚拟化产品市场。

2.2 VMware 的虚拟化产品

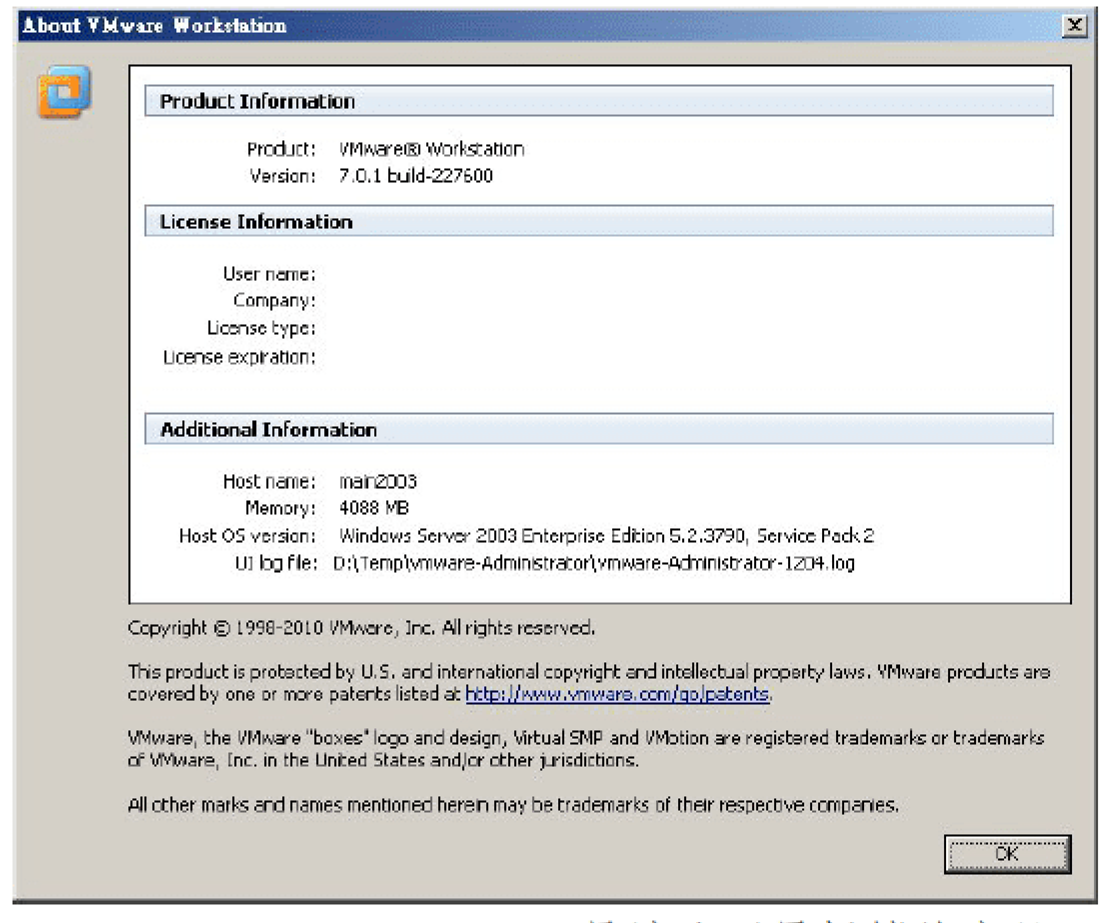
VMware 专注于虚拟化产品的开发，因此当前为止已经有数十种产品线，从最底层的 OS 虚拟化一直到最上层的应用软件虚拟化都有（请参见第 1 章的计算机的服务层级）。当然 VMware 也推出了在虚拟化过程中的辅助工具，如物理机转换虚拟机的 Converter 软件或是网页界面的成本 ROI 计算器。我们在这一小节就来看看几个比较重要的产品。

2.2.1 个人用产品（寄居架构）

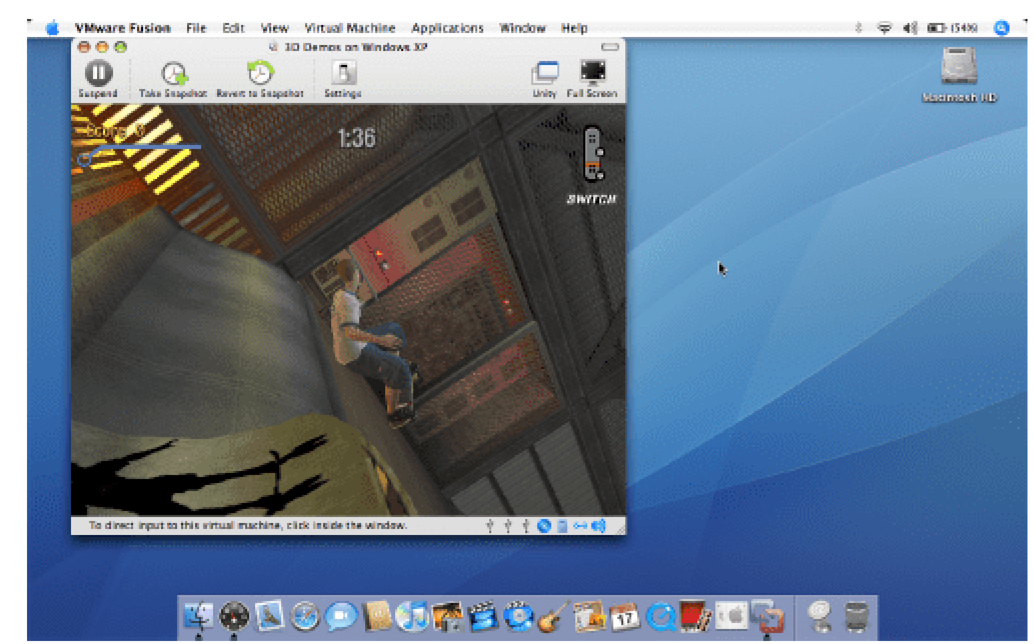
虽然本书的主角是 VMware vSphere，但不可讳言的，VMware 中最有名气的还是 VMware Workstation。但除了 Workstation 之外，VMware 在寄居架构上也有其他不错甚至免费的产品。

1. VMware Workstation

Workstation 是 VMware 的第一个产品，也是到目前为止销量最好的产品。Workstation 是标准的寄居架构产品，必须在另一个操作系统上运行 Hypervisor，但也因为是寄居架构，因此兼容性特别好，运行标准的 OS in OS 任务一点都不吃力。当前最新的版本是 7.01，拥有 Windows 和 Linux 版本，而在 Mac OS 下的 VMware Workstation 则称为 VMware Fusion，功能和 Workstation 完全相同。



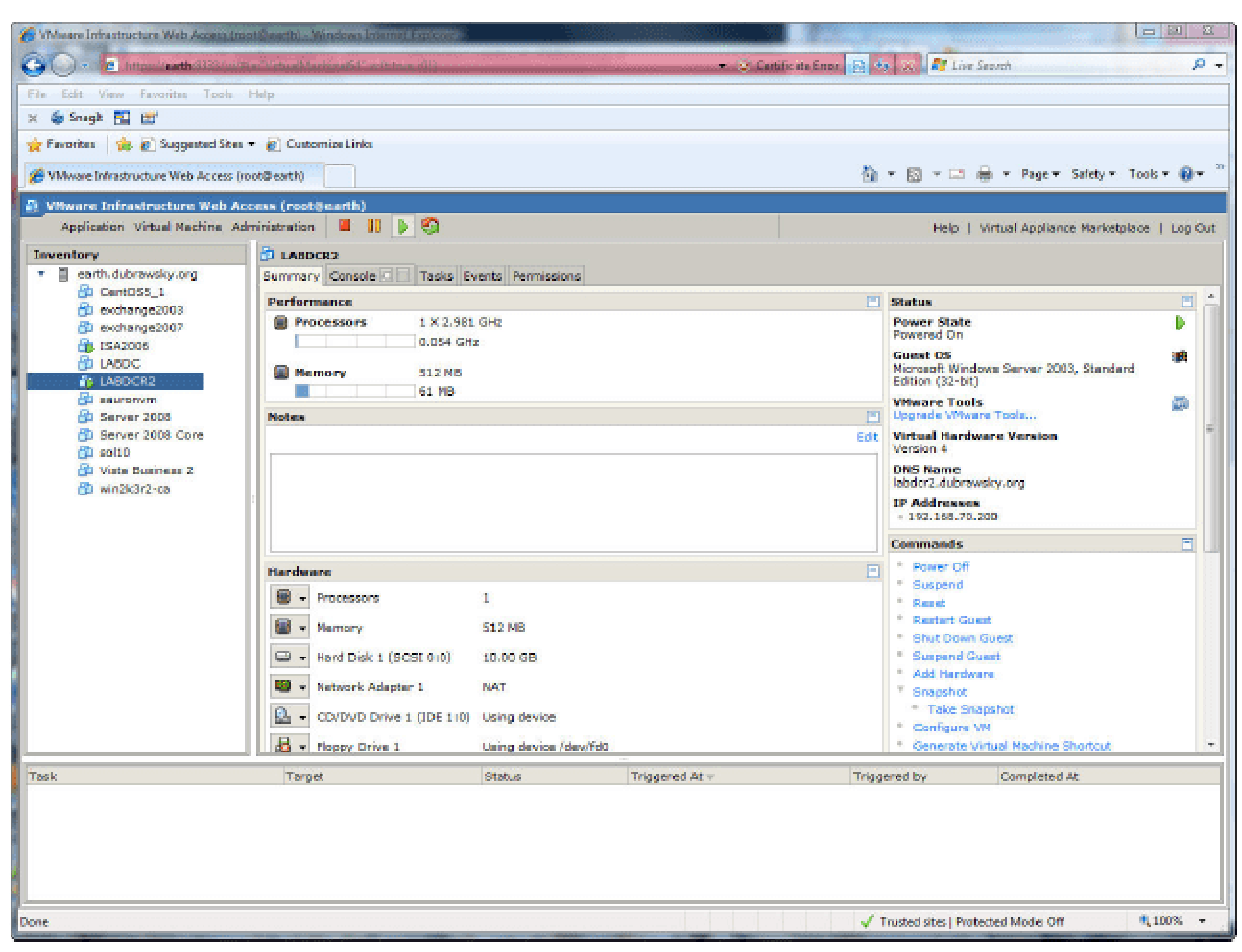
▲ VMware Workstation 是该公司最畅销的产品



▲ 在 Mac OS 下称为 VMware Fusion

2. VMware Server

微软在收购了 Virtual PC 之后，也推出了服务器专用的 Virtual Server，并且以免费的方式授权。Virtual Server 是以 Web 为界面的寄居架构产品，因此很快就攻占了 Windows 2003/2008 的市场。VMware 为了迎战 Virtual Server，也将从前的产品 GSX 变装为 VMware Server，并且也以免费的方式授权。VMware Server 的功能和 Workstation 比较一点也不差，但需要有 Web 服务器，如 IIS 或是 Apache/Tomcat，当前最新的版本为 2.0，并且可以在 Windows 或 Linux 下运行。不过这里要说明的是，虽然 VMware Server 是一个寄居架构的产品，但 VMware 还是将 VMware Server 归类于企业级而非个人用的产品。



▲　VMware Server 虽然是寄居架构，但也是服务器的产品

3. VMware Player

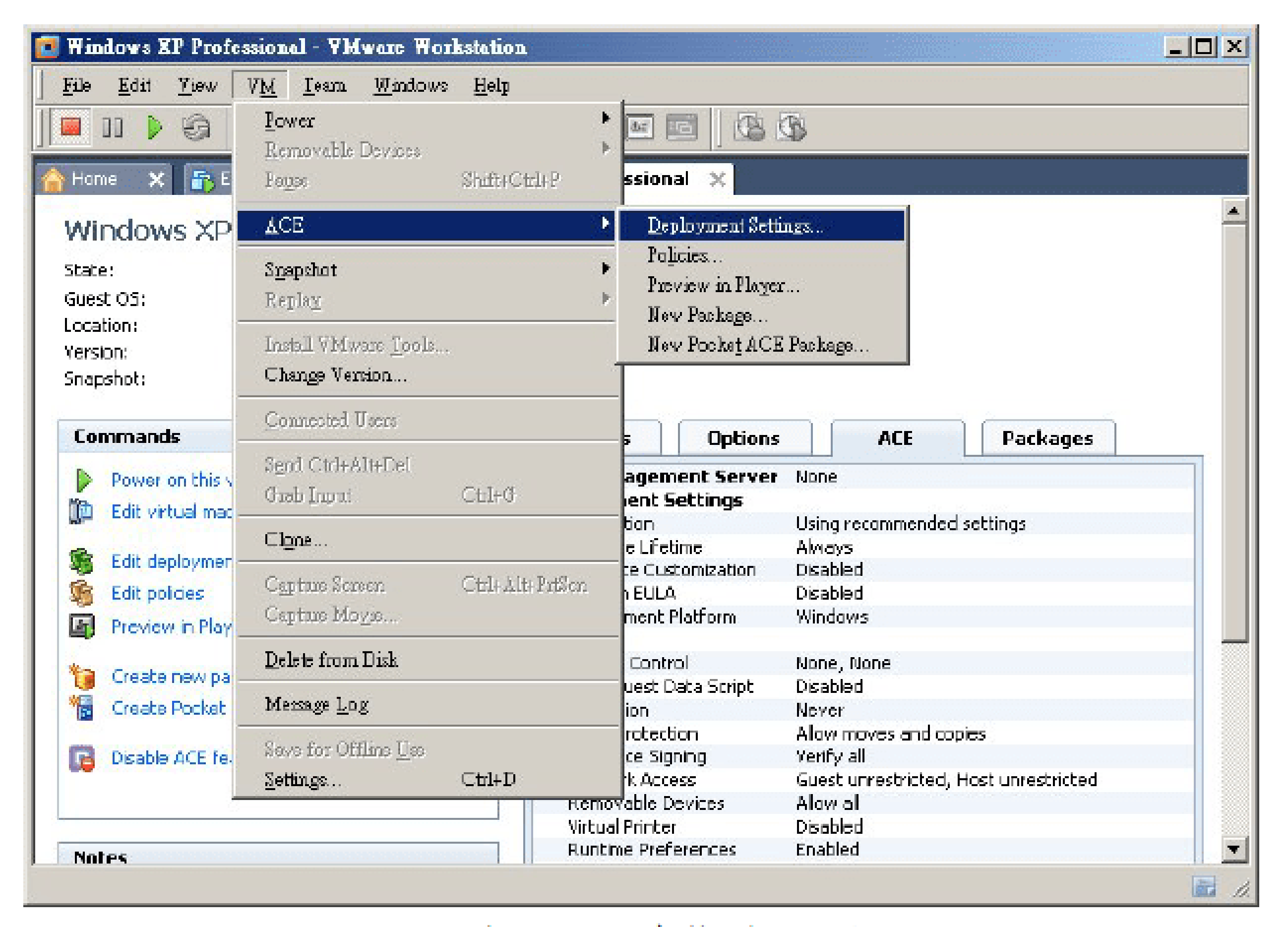
虚拟机的功能太强大，常常让人不知如何下手。如果你只想激活并创建简单的 OS in OS，功能越少效能反而越好。为了能让只需要基本需求的人也能使用 VMware 的产品，VMware 也推出了简单的 VMware Player。VMware Player 就是一个运行由 VMware Workstation/Server 创建的虚拟机的工具，当前最新的版本为 3.0，在 3.0 之后，也可以用来创建虚拟机了。



▲　图为 VMware Player

4. VMware ACE

在 Workstation 版本里有一个 VMware　ACE 功能，一般使用 Workstation 的人都很少用。事实上 ACE 功能就是针对在 Workstation 上运行的虚拟机的再细部配置，如权限、部署、在别台机器运行细项等。举例来说，你希望创建出来的 VM 不能被用户更改内存大小，就可以用 ACE 来配置，并且在发布 VM 后，所有用户都无法更改这个 VM 的内存大小。这一类的配置很多，但超出本书的范围，有兴趣的读者可以自行研究。



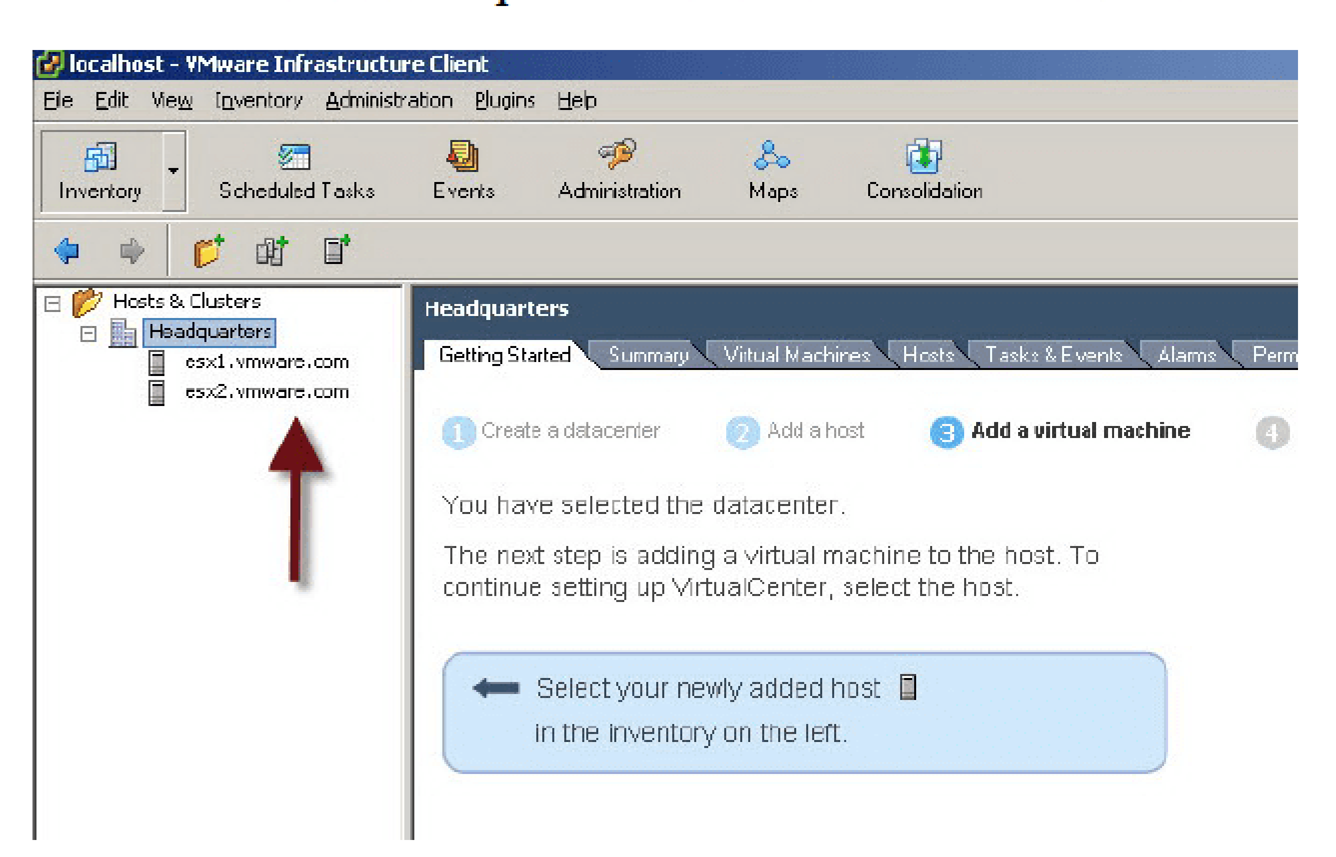
▲　图为 ACE 功能的配置画面

2.2.2　企业用产品（原生架构）

VMware 单独的原生架构产品不多，但最重要的就是 ESX/ESXi 两个服务器。这两个 Hypervisor 是组成整个 VMware vSphere 王国的内核，一般不单独出售，但也可以独立使用，下面就来看看 VMware 原生架构的产品。

1. VMware vCenter Server

VMware vSphere（以下简称 vSphere）是一个完整的解决专案而非单个产品，由硬件、Hypervisor、功能、服务、虚拟机等部件组成。整个 vSphere 的管理靠的是 VMware vCenter Server，没有 VMware vCenter Server 也就没有 vSphere，本书的热点也是围绕的 VMware vCenter Server。



▲　这是 vCenter Server，是 VMware vSphere 最重要的部件

2. VMware ESX Server

VMware ESX Server（以下简称 ESX）就是一个简单的原生架构 Hypervisor，在其上可以安装不同的 OS。在 ESX 之上，VMware vSphere 提供了许多功能，但如果不使用 VMware vSphere 而单独安装 ESX 也有许多强大的特色。VMware 将 ESX 加壳在不同的 VMware vSphere 产品中，本书接下来的章节会一一介绍。

```
[root@ESX3 volumes]# ls
4963a483-30104bc1-b0e1-000c29a80997  ESX3:storage1
[root@ESX3 volumes]# ls -l
total 1024
drwxr-xr-t    1 root     root          1260 Mar 10 22:17 4963a483-30104bc1-b0e1-0
00c29a80997
lrwxr-xr-x    1 root     root            35 Mar 11 10:42 ESX3:storage1 -> 4963a48
3-30104bc1-b0e1-000c29a80997
[root@ESX3 volumes]# _
```

▲ ESX Server 拥有 Linux 的配置画面

3. VMware ESXi Server

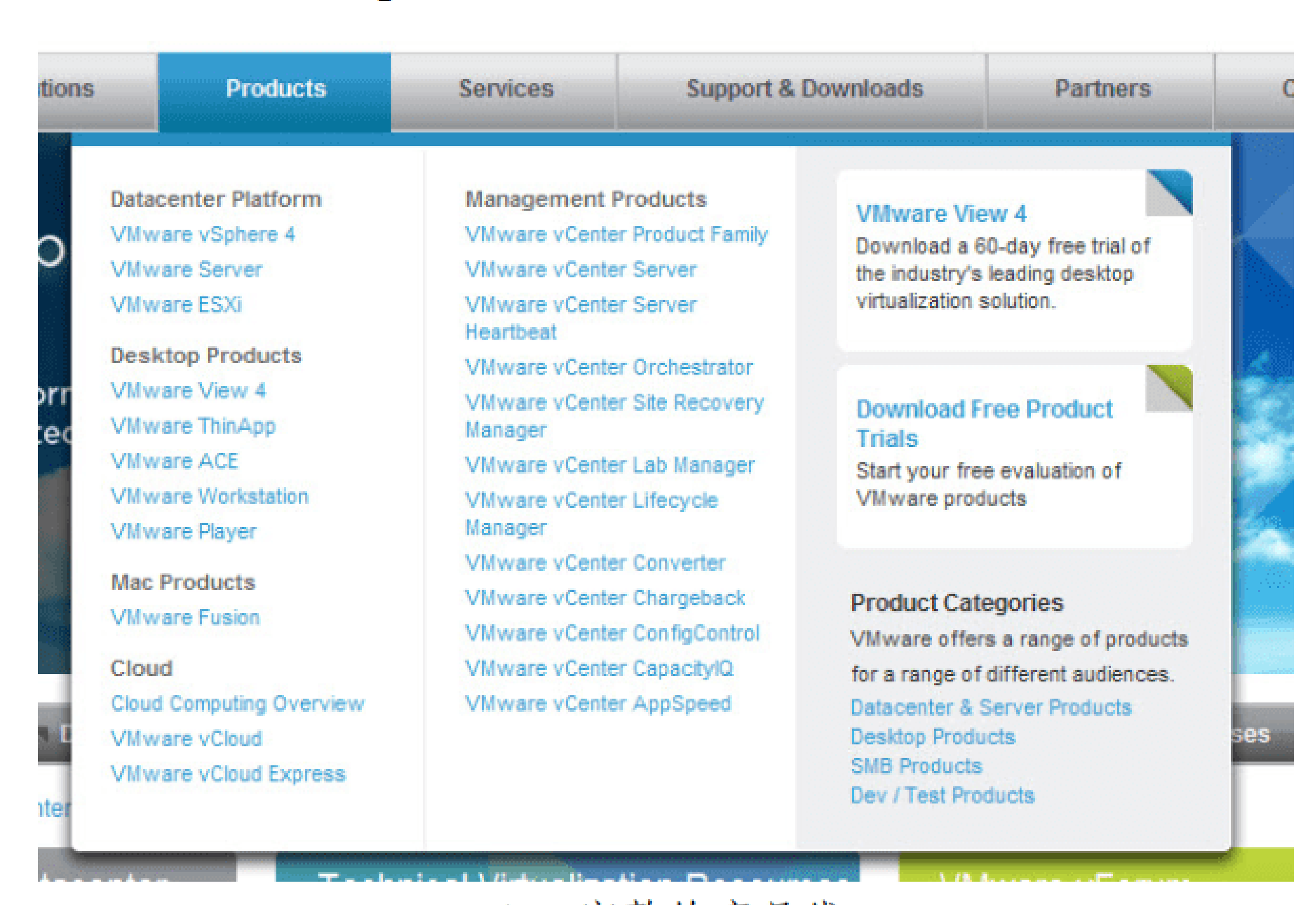
VMware ESXi Server（以下简称 ESXi）也是一个原生架构的 Hypervisor，但拿掉了很多管理用的界面，算是一个轻量级的 Hypervisor。ESXi 是笔者最推荐的原生架构 Hypervisor，在取得了授权之后，更可以和 ESX 集成至 VMware vSphere 中，更可以用 USB 或光盘引导，成为一个可携式的 Hypervisor。当前许多服务器厂家开始和 VMware 合作，将 ESXi 集成至系统的 BIOS 中，成为连硬盘都不需要的 Hypervisor，本书稍后也会有详细的介绍。



▲ ESXi 的界面简单，一定要通过 vCenter 管理

2.2.3 和虚拟机相关的其他产品

为了帮助企业进行虚拟化，VMware 也推出了一些辅助工具，这些辅助工具主要分为评估、测试、转换、管理几大方面。在此只介绍和本书主题相关的产品，如果读者想了解 VMware 所有产品线，可以到 www.vmware.com/products/检视。



▲ 完整的产品线

1. 评估类别

企业要上虚拟化之前，一定要经过适当的评估。VMware 使用了大量的评估工具让企业能安心转换平台，这些工具主要是用在安装 vSphere 之前确定兼容性或是成本的估量工具，读者在正式安装前可以下载使用，我们也会在本书稍后说明。

▶ 常用的 vSphere 评估工具

- vMark：评测 VM 中应用软件运行速度的软件。
- Capacity Planner：用来观察硬件使用效率的软件，可决定虚拟化的合并率。



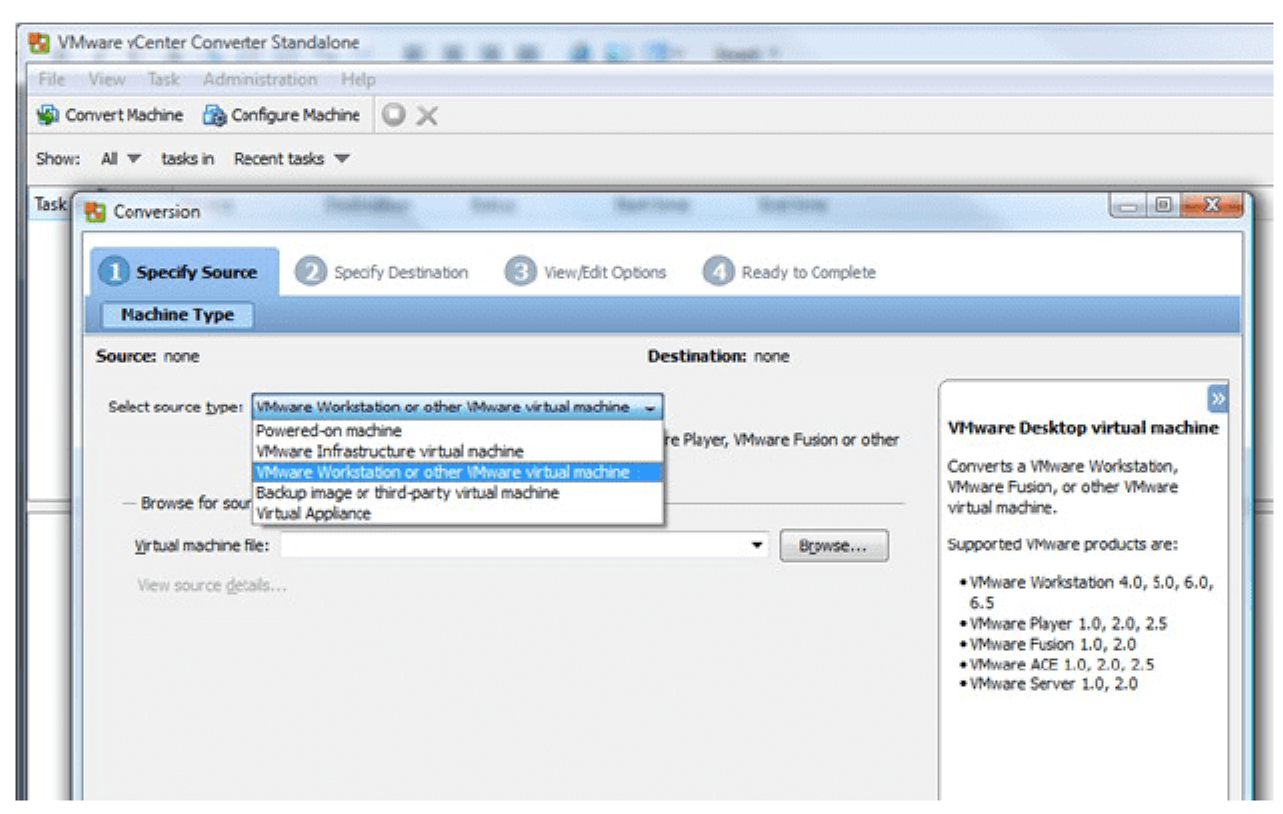
▲ Capacity Planner 可以帮助决定服务器的合并率

2. 管理类别

管理类的产品都以 VMware vCenter Server 为主，下面就是最常用的 VMware vCenter Server 管理功能介绍。

▶ VMware vCenter Server 的产品

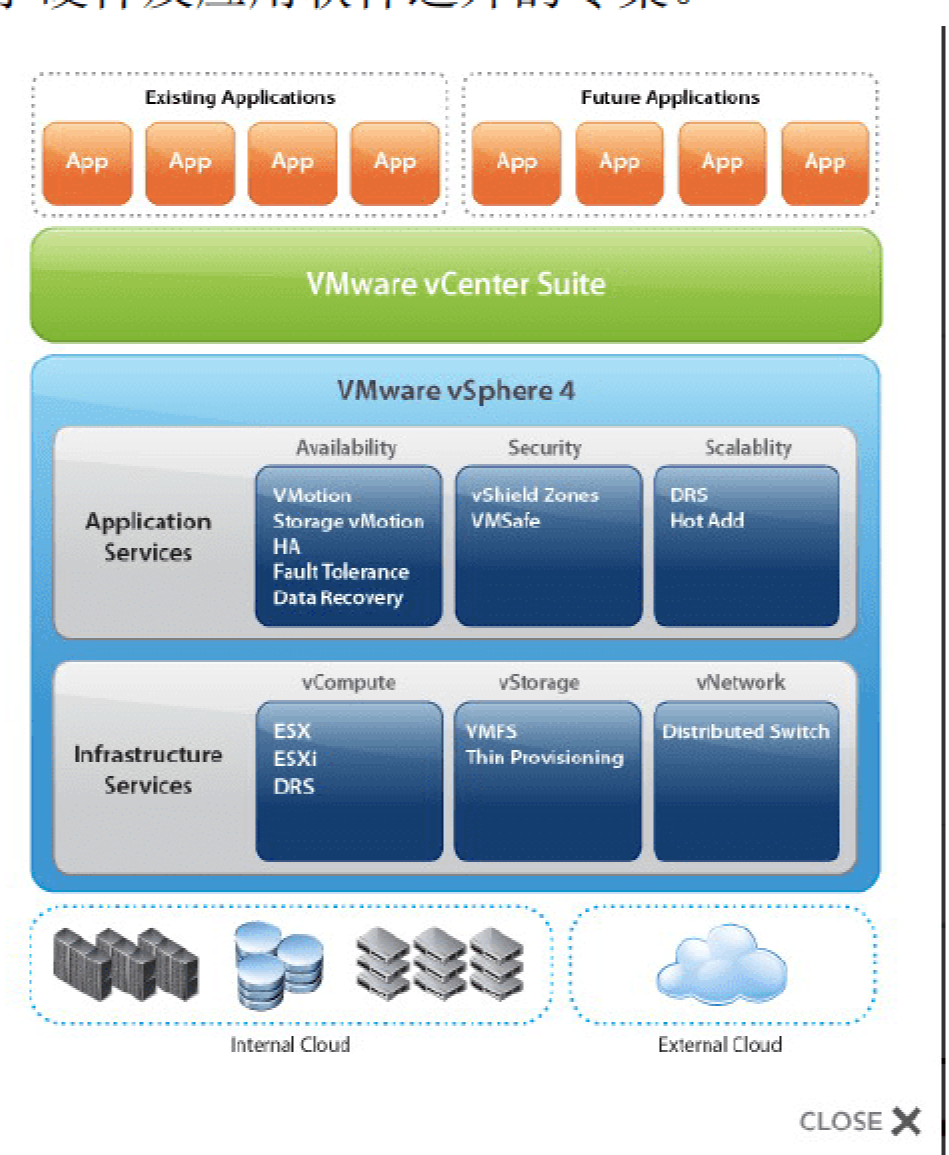
- Server Heartbeat：多台 vCenter Server 之间的可用性管理。
- Orchestrator：vSphere 自动化工具，可让许多常用的任务自动运行。
- Site Recovery Manager：当 vSphere 蓝屏时，可简化服务器回退的过程。
- Lab Manager：集成的用户管理中心，让大家都能访问相关信息。
- Lifecycle Manager：管理 VM 的生命周期工具。
- Converter：VM 最重要的工具，帮助企业将物理机转成 VM。
- Chargeback：管理和评估虚拟化的成本及花费。
- ConfigControl：可管理 vSphere 架构的完整配置。
- Capacity IQ：评估服务器合并或虚拟桌面的硬件能力。
- AppSpeed：在虚拟化之后，可知道应用软件在 VM 上的效能。



▲ Converter 是 P2V 中最常用的功能

2.3 完整的 vSphere 架构

vSphere 是一个完整的 IT 架构而非单个产品，可适用于各种规模的企业，其云端化的特色让大型企业更可以和第三方的资源结合，成为企业的单个入口解决专案。在进入本书的每一章节详细讲述使用方法之前，本节先对 vSphere 整体架构进行简单介绍，用下图来介绍 vSphere 的组成架构，VMware 提供的是除了硬件及应用软件之外的专案。



▲ 完整 vSphere 的架构

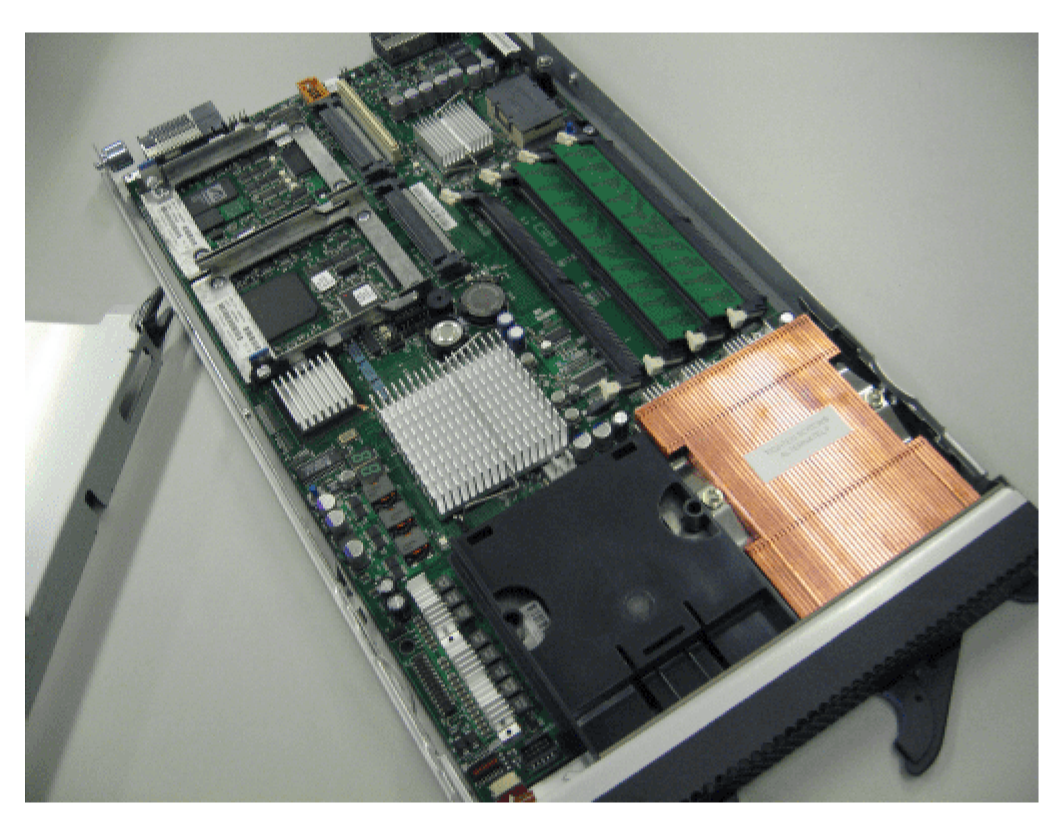
2.3.1 vSphere 的组成部件：云端部分

vSphere 将硬件资源定义成“云端”，这里所谓的云端是指平台及架构部分（PaaS 和 IaaS），分为内部云端和外部云端。

1. 内部云端

内部云端是由各种硬件资源所组成，并且由 vSphere 负责统合云端资源，在 IaaS 及 PaaS 中，

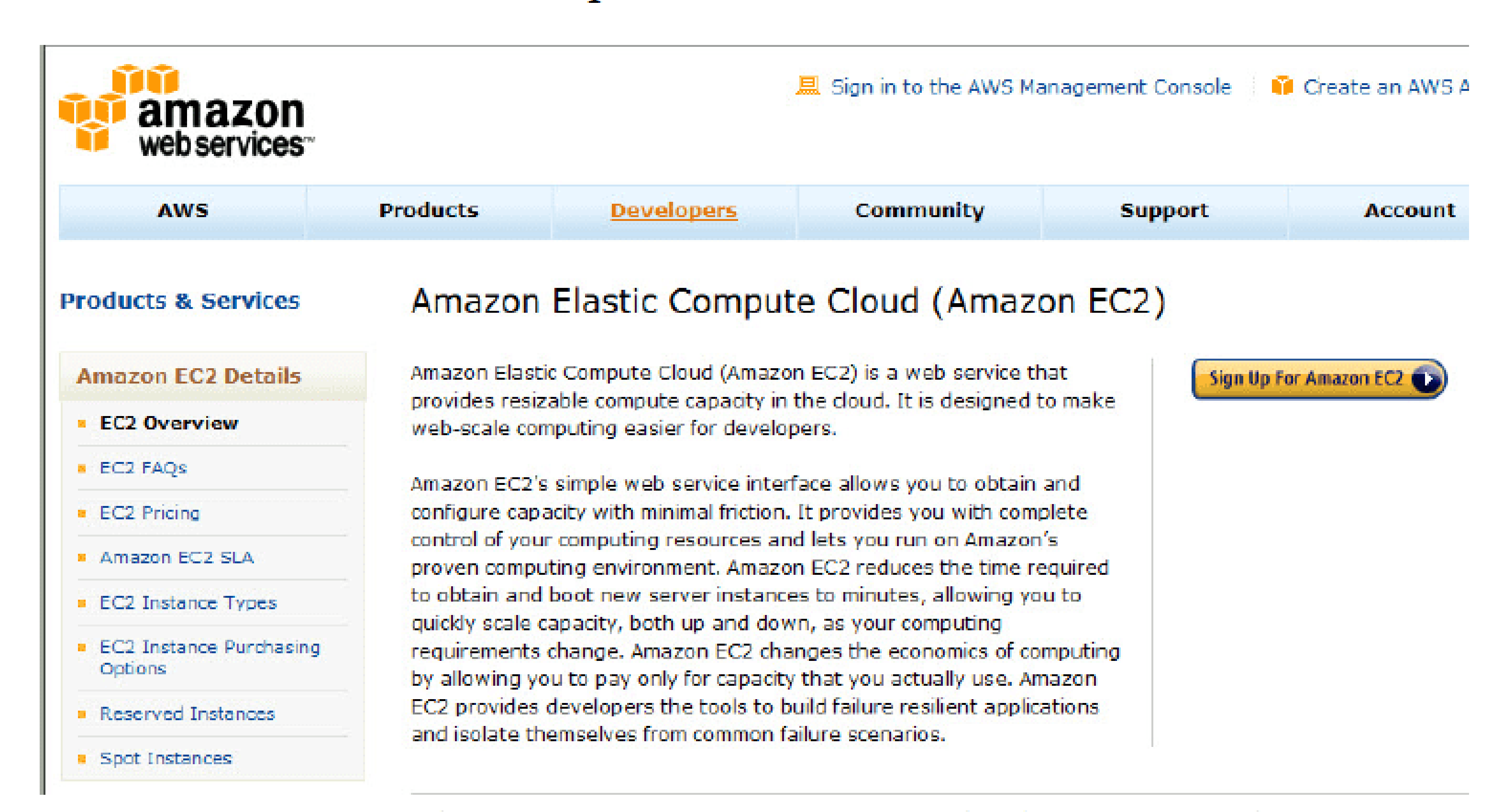
我们定义的资源为硬件及 OS 资源。硬件主要有 CPU 运算能力、RAM 以及存储空间，而 PaaS 则是有各式各样的操作系统。



▲ 机房内的服务器就是标准的内部云端

2. 外部云端

当前许多第三方厂家提供了各式各样的 IaaS 及 PaaS，包括 Amazon 的 EC2 或是 VMware 的 vCloud Express。vSphere 可以将这些资源集成到企业的的 IT 架构中，使用手段可以是通过第三方厂家提供的 API、Web Service 或是 vSphere 提供的 API。



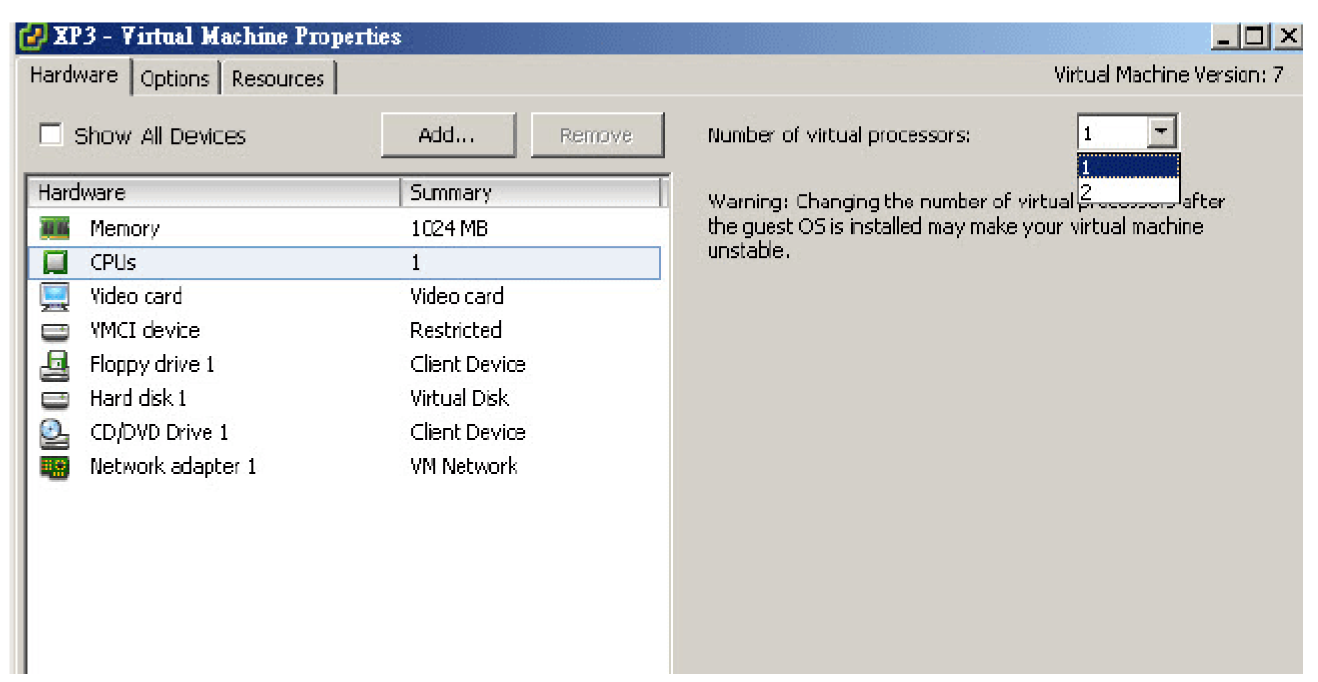
▲ 已经被谈到烂的 Amazon EC2 是标准的外部云端

2.3.2 vSphere 的底层：架构服务（Infrastructure Service）

有了 CPU/RAM/存储之后，我们需要一个 Hypervisor 将资源集成，并且向上提供一个虚拟的硬件资源给 VM 使用，这部分最重要的就是 ESX 和 ESXi 服务器负责将硬件资源虚拟化，其他部件用来向上“欺骗”VM，主要分为运算部分的 vCompute、存储部分的 vStorage 以及网络部分的 vNetwork。

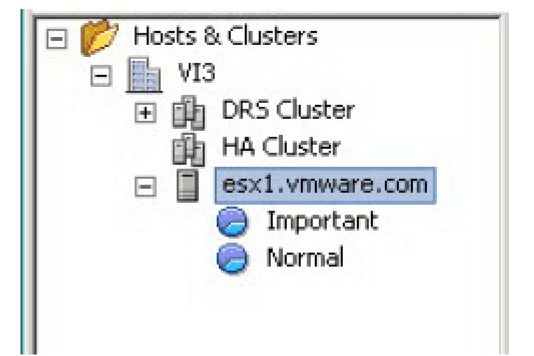
1. vCompute 部分

VCompute 包括了 ESX/ESXi 以及 DRS。ESX/ESXi 就是安装在物理服务器上的 Hypervisor，功能就是将 CPU 资源累加之后分配；分配内存以及存储空间的规划。ESX/ESXi 服务器是 vSphere 的灵魂。在没有 vSphere 整体架构的情况下，ESX/ESXi 也能独立运行，单纯当一个 Hypervisor，但大部分企业级功能就无法使用了。



▲ 最重要的部分是 vCPU

分布式资源调度（Distributed Resource Scheduler，DRS）也属vCompute，主要功能就是让 VM 能自己"找"最适合的物理服务器。举例来说，一个运行 SQL 的 VM 原来运行顺利，突然负担变大时，这个 VM 会在 vSphere 的集群中找一台 CPU 更强、RAM 更多的物理服务器移过去，本书会有一个完整的章节介绍这个功能。

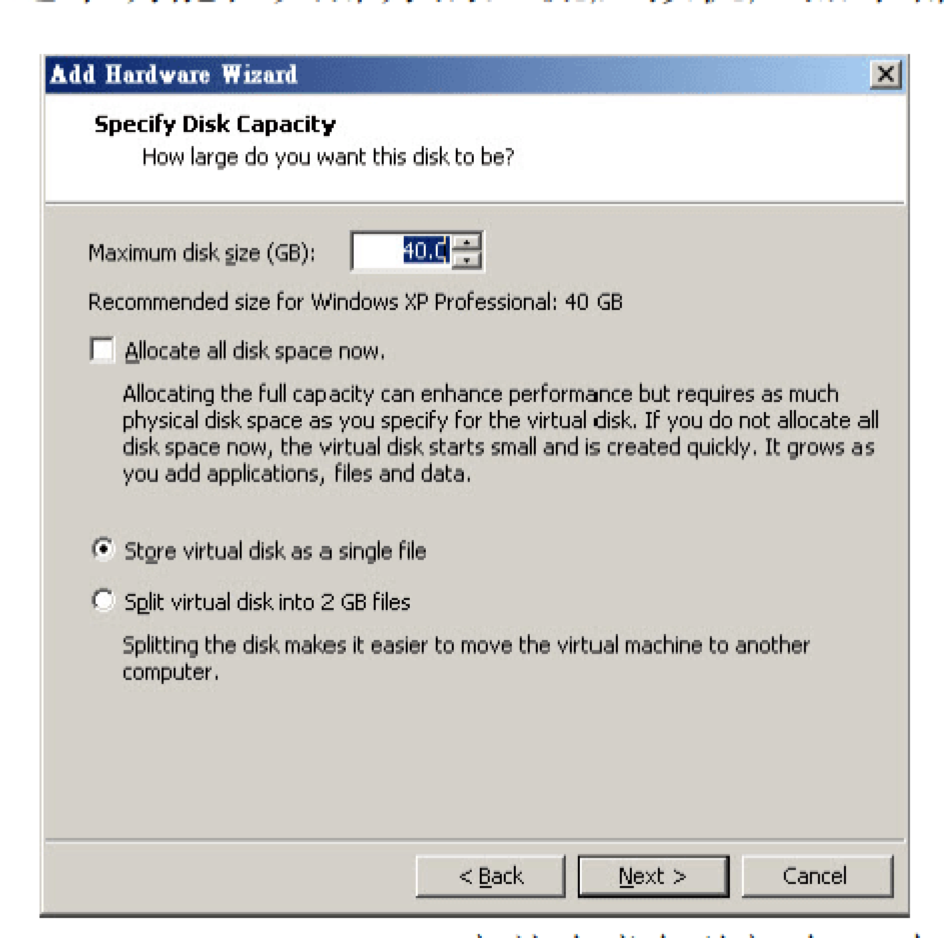


▲ DRS 可创建永不蓝屏的环境

2. vStorage 部分

vStorage 包括 VM 所在硬盘的文件系统 VMFS 以及动态分配大小的 Thin Provisioning。VMFS 是放置 VM 的文件系统，就像 Windows 下的 NTFS 或 Linux 下的 EXT3/4 一样，是所有 VM 文件的基本存储空间。

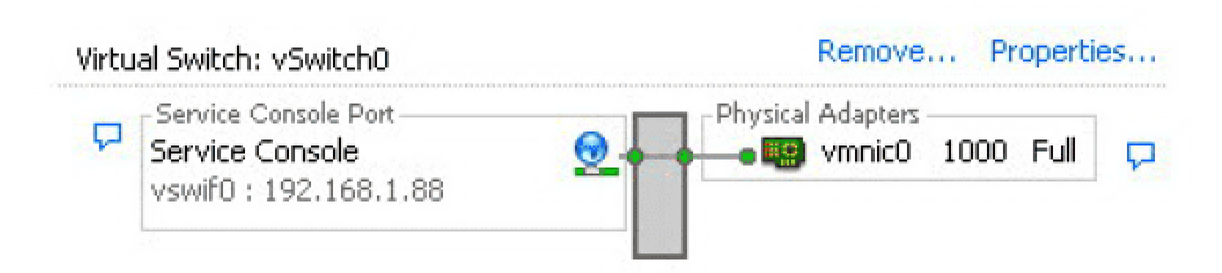
Thin Provisioning 的观念很简单，因为一般使用硬盘是不可能一安装 OS 就把硬盘装满的。Thin Provisioning 可以动态增大硬盘，当需要容量时才会配置真正的空间，虽然会影响速度，但可以真正省下硬盘的空间，这个功能在大部分的主流虚拟机产品中都有。



▲ 其实在 VMware Workstation 中就有类似的概念，动态分配磁盘

3. vNetwork 部分

VM 的网络是最复杂的。一台物理服务器上可能有很多个 VM，每一个 VM 都可以和物理服务器处于同一个网段（举例来说，都是 192.168.1.0/24），也可以使用 NAT 或是 VLAN。这么复杂的网络架构，可能只通过一个物理的网卡，因此提供这么强大的网络功能，vSphere 的 vNetwork 是十分重要的。



▲ VMware 的网络是热点之一

vSphere 提供一个称为 Distributed Network 的架构，不但有完整的 Bridged/NAT/Host only 架构，更和 Cisco 合作推出了一个专门安装在 vSphere 上的分布式网络交换机，本书也会花很大篇幅说明来 vNetwork 的原理和实践。

2.3.3 vSphere 的底层：Application Service

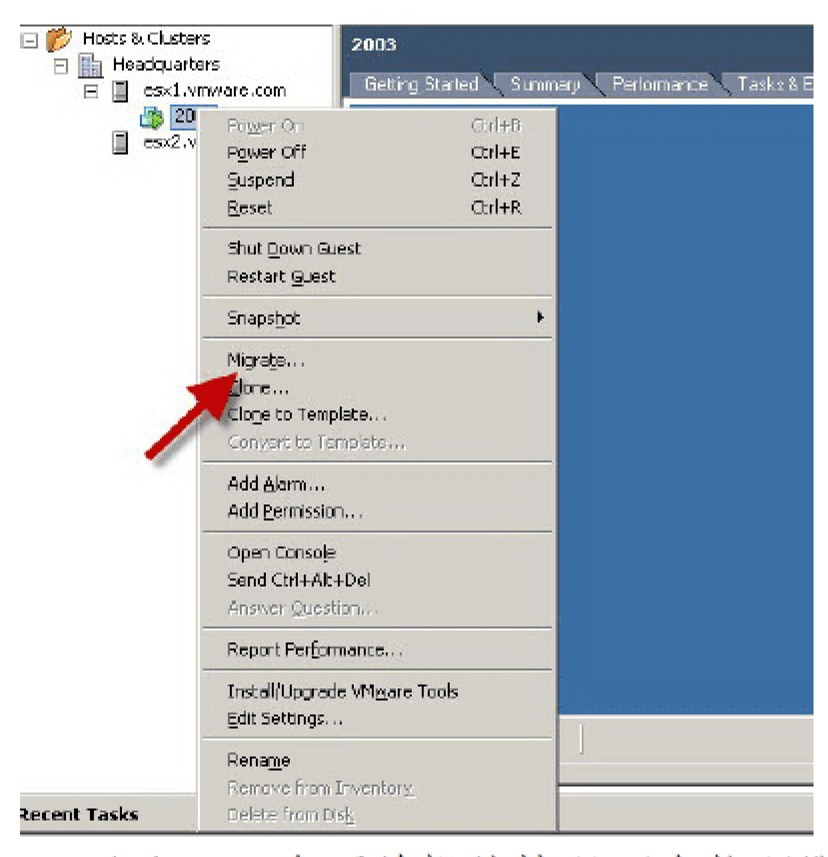
应用软件服务是针对 VM 的，可以让多台服务器多个 VM 排列组合，达成企业应用的目的。大部分的企业在使用 vSphere，多半都是以其的功能为主。

1. 可用性（Availability）

所谓的可用性，就是企业的服务永远不会中断，不管是服务器蓝屏或是应用软件蓝屏，都不影响用户对服务的访问。vSphere 在这一部分提供的功能有下面几个：

▶ **可用性的功能列表**

- VMotion：VM 动态转移，可以把 VM 从一台物理服务器移到另一台上。
- Storage VMotion：VM 磁盘动态转移，可以把 VM 硬盘从一个存储设备移到另一台上。
- HA（高可用性）：VM 会在一台服务器蓝屏后，移到另一台上，服务永远不会中断。
- 冗余（Fault Tolerance）：随时有一个动态的服务器待命，当有一台服务器蓝屏时，不需要 HA 或 VMotion 就立即接手。
- Data Recovery：服务器或 VM 蓝屏之后的数据回退功能。



▲ VMotion 可以让 VM 在不关机的情况下在物理服务器上移动

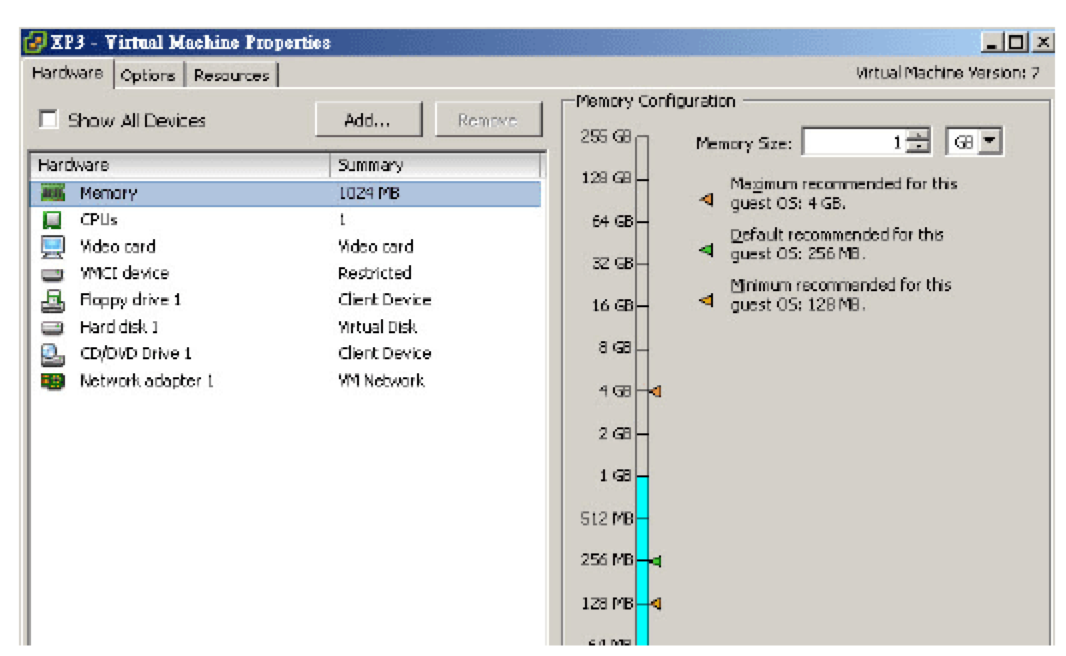
2. 安全性（Security）

安全性包括了 vShields Zones 和 VMSafe 两部分。在操作虚拟机时，网卡一定要将这些虚拟机视为物理机，因为它们也有安全上的漏洞，黑客攻击时，不会管你是虚拟机还是物理机，因此虚拟机的网络安全很重要。在 VMware 中有 vShield Zones 和 VMSafe，可以让物理机直接上不同的虚拟机，甚至是不同物理机上的不同虚拟机之间的关联，不需要通过外界的防火墙或是路由器

获取监控。当企业虚拟化进入管理为王的时代，这些管理为主的功能将决定虚拟机市场的胜负。稍后将会有完整介绍。

3. 可扩展性（Scalability）

再强大的系统也有遇到效能瓶颈的一天，从前的计算机流行纵向升级（Scale up），就是在同一台电脑上安装更多更快的 CPU，加装更多的 RAM。但纵向升级的容量有限，一台电脑不可能有无限个 CPU 和内存插槽，因此横向升级（Scale out），就是用更多的物理服务器来添加效能。vSphere 提供了 DRS 和 Hot Add，让 VM 能动态转移到更快捷的物理服务器上，而 Hot Add 的功能可以让 VM 在不关机的情况下直接添加 vCPU 或内存，这对系统的高可用性也有极大的帮助。



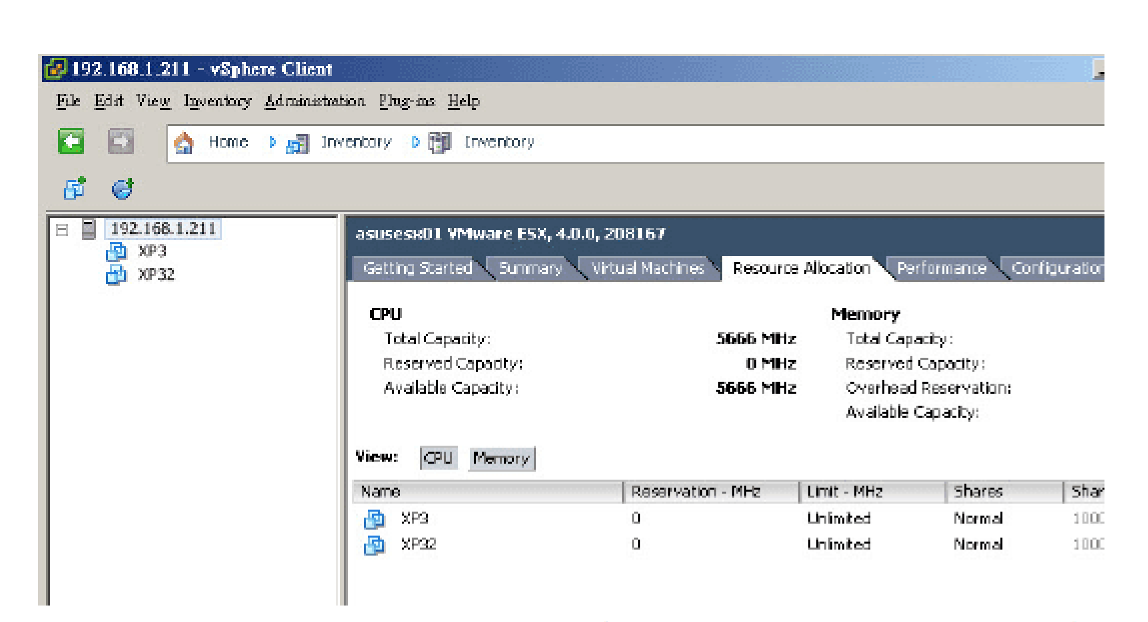
▲ 在 VM 中，内存一般是一对一的对应

2.3.4 vSphere 的神经中枢：VMware vCenter

vSphere 这么强大的功能，如果没有一个中央总控的管理界面，是无法应付日趋复杂的 IT 架构的。vSphere 提供了 vCenter，是 vSphere 的神经中枢，本书所有的配置及功能都以 vCenter 为主。

1. VMware vCenter Client

这是一个 Windows 端的实用程序，用来直接总控单台的 ESX/ESXi。在 vSphere 中，所有的 VM 管理、创建、运行、维护都靠 vCenter Client。理解 vSphere 的基本 VM 其实就是理解 vCenter Client 的操作。



▲ vCenter Client 是操作单个 ESX/ESXi 的客户端

2. VMware vCenter Server

vCenter Server 大概是 VMware 中功能最复杂的产品了。前面所提的云端、架构、应用软件等，都要靠 vCenter Server 来落实。vCenter Server 的安装十分巨大，需要有 DNS、AD 以及 SQL Server，使用 vCenter Server 必须对 Windows 及 Linux 系统有很深入的理解，不过读者们可以照着本书的步骤一步步来，就可以完全掌握 vCenter Server 了。



▲ 管理多台 ESX 则需要 vCenter Server，但仍然是用 Client 端登录

2.3.5　最重要的部件：虚拟机

虚拟机（Virtual Machine，VM）是真正运行功能的地方，一个 VM 可以被视为一台物理机器，拥有自身的硬件资源（虽然是被虚拟化过的）。在 vSphere 中，一台 VM 通常由配置文件 VMX 以及虚拟硬盘 VMDK 组成。

1. VMX 文件

当在 vSphere 上激活一台 VM 时，事实上就是 Hypervisor 服务可读一个 VMX 文件，并且依照 VMX 文件中的指示来配置物理服务器的硬件资源。VMX 文件是个文本档，在创建 VM 时可写的，你也可以手动更改，但通常是通过 vCenter 来做。



```
.encoding = "Big5"
config.version = "8"
virtualHW.version = "7"
scsi0.present = "TRUE"
scsi0.virtualDev = "lsilogic"
memsize = "512"
scsi0:0.present = "TRUE"
scsi0:0.fileName = "Ubuntu-000004.vmdk"
ide1:0.present = "TRUE"
ide1:0.fileName = "auto detect"
ide1:0.deviceType = "cdrom-raw"
ethernet0.present = "TRUE"
ethernet0.wakeOnPcktRcv = "FALSE"
pciBridge0.present = "TRUE"
pciBridge4.present = "TRUE"
pciBridge4.virtualDev = "pcieRootPort"
pciBridge4.functions = "8"
pciBridge5.present = "TRUE"
pciBridge5.virtualDev = "pcieRootPort"
pciBridge5.functions = "8"
pciBridge6.present = "TRUE"
pciBridge6.virtualDev = "pcieRootPort"
pciBridge6.functions = "8"
pciBridge7.present = "TRUE"
pciBridge7.virtualDev = "pcieRootPort"
pciBridge7.functions = "8"
vmci0.present = "TRUE"
roamingVM.exitBehavior = "go"
displayName = "Ubuntu"
guestOS = "ubuntu"
nvram = "Ubuntu.nvram"
virtualHW.productCompatibility = "hosted"
ft.secondary0.enabled = "TRUE"
tools.upgrade.policy = "useGlobal"

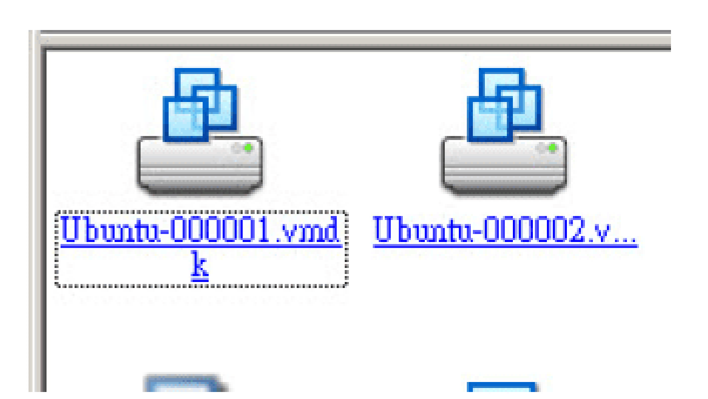
extendedConfigFile = "Ubuntu.vmxf"

ethernet0.addressType = "generated"
uuid.location = "56 4d 21 1f b4 41 e8 96-d4 ca 50 e7 c6 4b e5 bd"
```

▲ 一个 VMX 文件是标准的纯文本

2. VMDK 文件

vSphere 中的硬盘配置有很多种，但不论是哪一种，都有一个 VMDK 文件来代表 VM 的硬盘。VMDK 文件通常置放在 VMFS 文件系统上，并且由 VMX 来给定可读。一般来说，我们在做物理机和虚拟机之间的转换时（Physical to Virtual，P2V），就是将物理机的硬盘数据映射成一个 VMDK 文件。



▲ 一个 VMDK 就是一个 VM 的硬盘

2.4　理解 vSphere 的不同版本

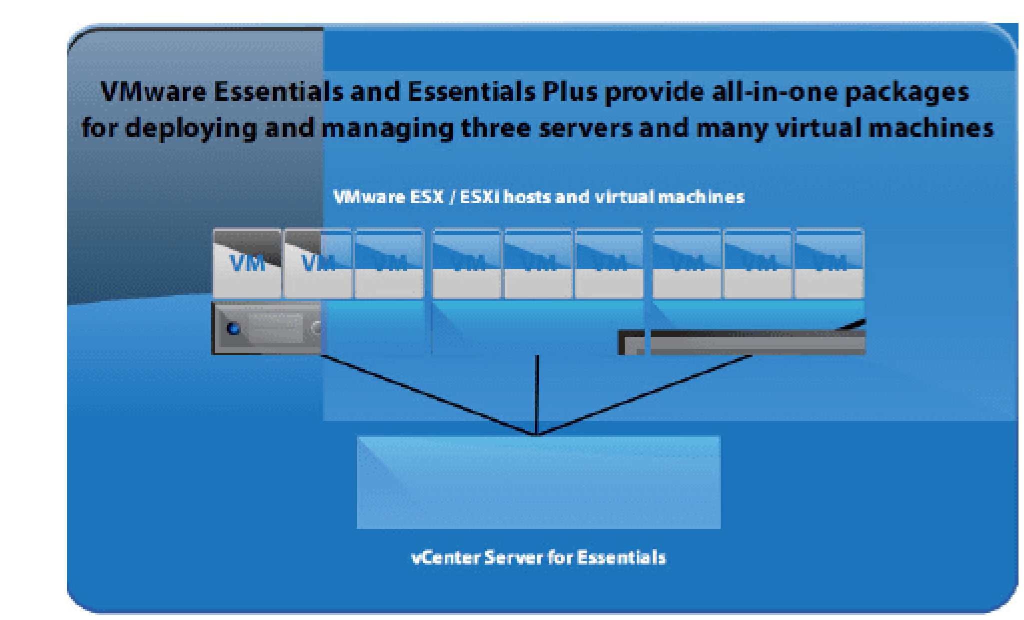
不同规模的企业对 vSphere 的功能需求也不同。10 人以下的小公司可能只有一个网页服务，只需要 ESXi 就可以落实，但云端机房可能有上万台服务器，需要的功能就完全不同。vSphere 对应到不同规模的企业，自然在定价和定位上有不同的针对性，我们在这一小节，就来看看 vSphere 的不同版本别。

2.4.1 小型企业用的版本

在针对人数较少的小型企业，VMware 的 vSphere 也推出了规约化的产品，这些产品的特色就是将不需要的功能卸除，并且将安装 vSphere 时需要的各种第三方组件（如 SQL、AD）都集成起来了。中小企业如果在预算有限的情况下，使用规约版本的 vSphere 是再适合不过的了。vSphere 规约版本主要有四个产品，分别是：

▶ 小型企业用 vSphere 的版本

- VMware vSphere Essentials
- VMware vSphere Essentials Plus
- VMware vSphere Essentials for Retail and Branch Offices
- VMware vSphere Essentials Plus for Retail and Branch Offices



▲ Essentials 是一个完全集成的专案，可以直接上线使用

1. 适用的对象

如果企业只有几台服务器，但也希望能有完整的虚拟化以及安全防护，那么使用 vSphere Essentials 是最好的选择。下面为建议购买的企业类型。

▶ 适合 vSphere Essentials 版本的企业

- 员工 10～50 人规模。
- 总服务器在 20 台以下。
- 有多个据点（Site），但每个据点都只有数台服务器。
- 不需要云计算。
- IT 人员 3 人以下。
- 需要快捷布建虚拟化。
- 服务的效能不是主要考虑热点。

2. Essentials 版本的内核部件

中小企业使用的 vSphere 并没有太多防灾或副本的功能，主要提供的是服务器合并以及管理的功能，主要有下面几个部件。

▶ 中小企业 vSphere 版本的基本功能

- Hypervisor Server：ESX 或 ESXi。
- Patch Manager：补丁管理员，掌管 Hypervisor 的升级及补丁。
- Management Agent：管理界面，联机至 vCenter Server Agent 程序。
- Management Server：管理服务器，使用 vCenter Server 的规约版。
- VM Storage Management：即为 Thin Provisioning。

这些版本的 vSphere 虽然功能不见得多，但对于小型企业来说应该是最完美的专案，因为多余的功能只会添加系统的复杂性和购置成本。而 vSphere 提供的内核部件，足以应付中小企业虚拟化的要求了。

3. Essentials 基本功能

这四个版本的 vSphere 提供了快捷的布建功能，让小型企业在购买之后能在最快的时间内落实虚拟化的架构。Essentials 包括了下列部件。

▶ Essentials 套装内核部件

- 多合一的授权，不需要针对个别功能买授权。
- 可安装在最多三台物理服务器上。
- 每台物理服务器最多只能有两个 CPU 插槽。
- 一台中央管理服务器（安装 vCenter Server）。
- ESX 或 ESXi（看具体安装的需要）。
- VMware vStorage VMFS（可安装在独立存储或 ESX 的本地硬盘）。
- 四路的虚拟 SMP。
- VMware vCenter Server Agent。
- VMware 的 vStorage API 使用。
- VMware 副本程序（VMware Consolidate Backup，以下简称 VCB）。
- VMware vCenter Update Manager，升级管理员。
- vCenter Server for Essentials。
- 一年的 Subscription。
- 技术支持为选购条目，并且视支持状况而定。
- 定价为 995 美元（此为引用售价，以当地具体售价为准）。
- Essentials for Retail and Branch Offices 功能和 Essentials 完全相同，但一次需至少购买十套。
- Essentials for Retail and Branch Offices 技术支持及 Subscription（SNS）需另外购买，最少一年。

▶ Essentials Plus 套装内核部件

- VMware Essentials 的所有功能。
- VMware HA 高可用性。
- VMware Data Recovery 数据回退。
- 技术支持及 Subscription（SNS）需另外购买，最少一年。
- 定价为 2995 美元（此为引用售价，以当地具体售价为准）。
- Essentials Plus for Retail and Branch Offices 功能和 Essentials 完全相同，但一次需至少购买十套。
- Essentials Plus for Retail and Branch Offices 技术支持及 Subscription（SNS）需另外购买，最少一年。

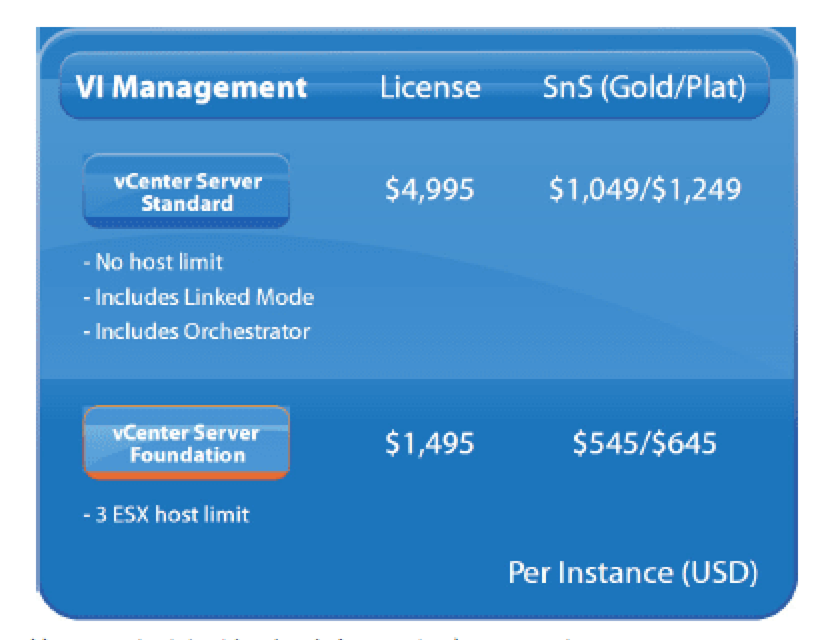
4. 有关 Essentials 的 vCenter

事实上在安装 vSphere 时，还需要安装 DNS、AD 域以及 SQL 才能管理 vCenter。为了让中小企业跳过这些麻烦，Essentials 特别在 vCenter Essentials 中加入了这些功能，使得 Essentials 竟然成为最完整的 vSphere 套件。其中包括了：

▶ vCenter Essentials 的部件

- Management Server：就是管理 VM 的 Server。
- Database Server：数据库服务器，有了这个就不用安装巨大的 MS SQL/Oracle。
- Search Engine：搜索引擎，让管理员能在 vSphere 中查找对象。
- vCenter Client：连入 vCenter Server 的客户端，也是管理 ESX/ESXi 的工具。
- VMware Web Access：使用 Web 连入 vCenter Server 的附加部件。

- VMware vCenter APIs：包括了.NET Framework 的扩展部件，可以让用户自行开发出需要的套件。



▲ Essentials 使用的是基本版三台机器的 vCenter Server Foundation

5. 成本精算

在理解 Essentials Plus 的功能之后，企业主最关注的就是购置成本。Essentials 的套件包括了所有 vSphere 所需的部件，因此用户不需要再另外购买任何额外的第三方套件，如 SQL Server，因此成本的计算很单纯，下表是成本的精算。

表 1 Essentials 的成本精算表

此为引用售价，以当地具体售价为准	纯 ESXi	Essentials	Essentials Plus	Essentials for Retail and Branch Offices	Essentials Plus for Retail and Branch Offices
Package	免费	$995	$2 995	$995×10	$2 995×10
金牌或白金 SNS	$545 $645	$545 $645	$545 $645	$545 $645	$545 $645
成　本	$545/645	$1 540/1640	$3 540/3 640	$10 495/10 595	$30 495/30 595

对中小企业来说，如果只有几台服务器，一个点的话，使用免费的 ESXi，然后购买一个一年的技术支持只需要 2000 元左右，平均一个月才一百多，就可以让公司成为虚拟化，每年省下来的电费和冷气费用都不止了。如果有多个点，需要购买 10 人版的，总价约 10 万元。这只是一台服务器的成本，对于虚拟化来说的确是很值得。

> **注意**
>
> **Essentials 的定价原则**
>
> VMware Essentials 的定价原则是以服务器为准（Instance）。一套软件可安装的最大单位是服务器，举例来说，Essentials 版本能安装三台物理服务器，如果有五台或六台，就得买两套安装，这和后面的中大型企业以 CPU 插槽数计价是不同的，读者们要特别注意。

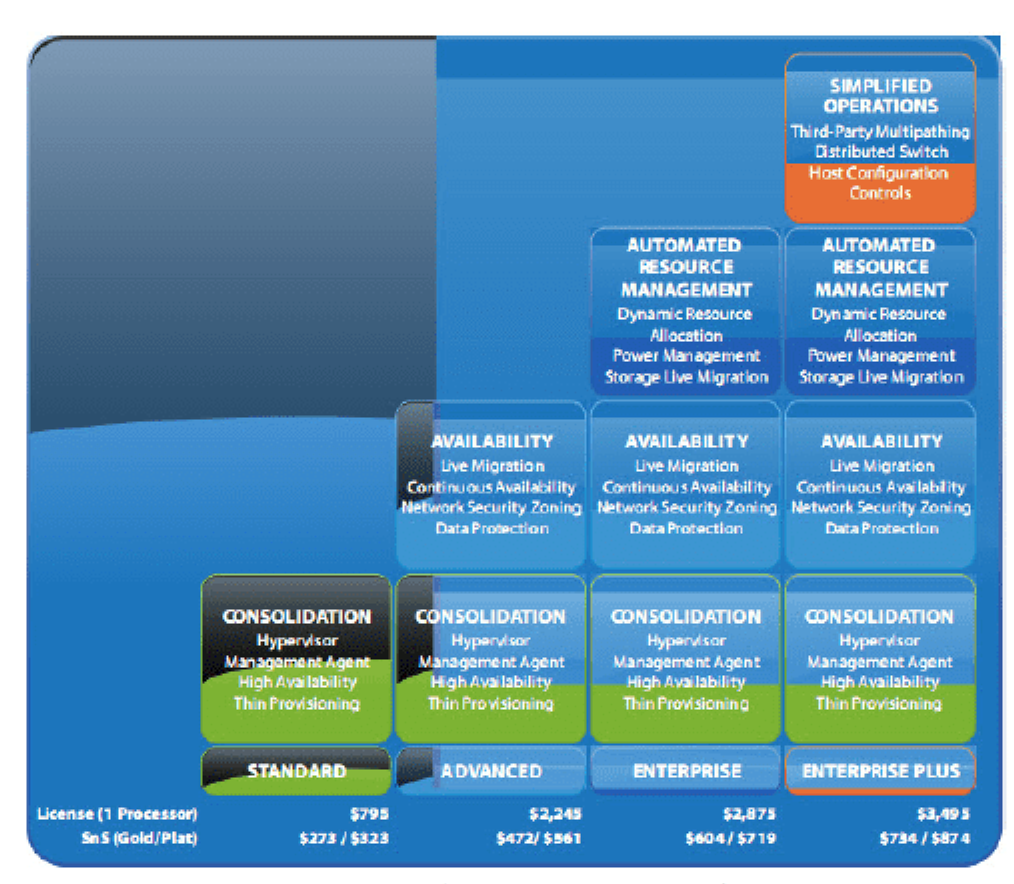
2.4.2 中大型企业的版本

大部分人关注的还是中大型企业用的 VMware vSphere，而 VMware 也将重心放在这一块，

本书所介绍的功能大部分也集中在这部分。中大型企业使用的 vSphere 版本也分成四个，分别为：

▶ **VMware vSphere 中大型企业版本**

- 标准（VMware vSphere Standard）
- 进阶（VMware vSphere Advanced）
- 企业（VMware vSphere Enterprise）
- 企业加强版（VMware vSphere Enterprise Plus）

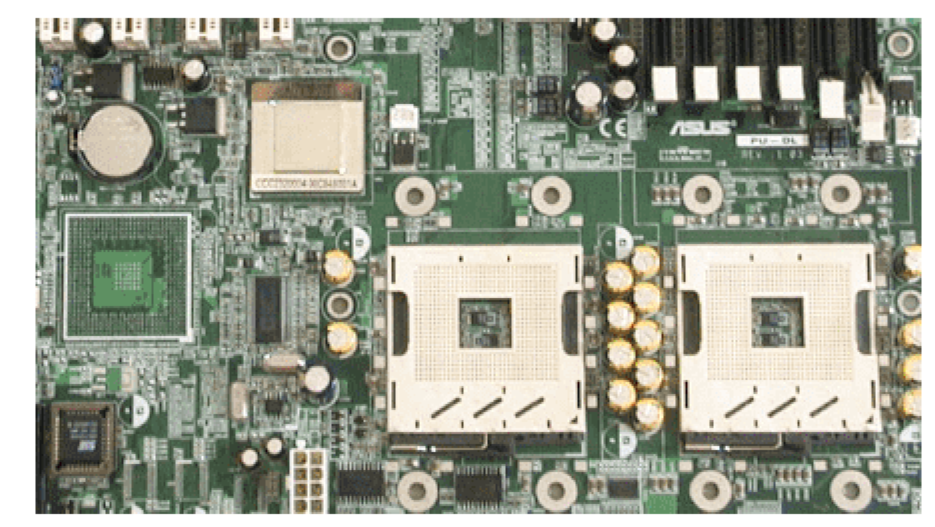


▲ vSphere 中大型企业版本的功能及成本图

这些版本的适用对象就是中大型企业，任何想要进入虚拟化领域的中大型企业，可以视需要的功能来购买。一般来说，公司有数十台以上的服务器，并且拥有自身的机房，通常都会需要使用中大企业版本的 vSphere。

1. 注意计价原则

特别要注意的是，VMware vSphere 中大型企业版本的计价是以 CPU 插槽为单位，只要每一个插槽的 CPU 是六内核以下就可以。举例来说，如果购买的服务器是 4WAY 的话，就必须购买四个授权才能合法使用。在进阶及企业加强版中，单个插槽的 CPU 可以拥有 12 个内核，有关 CPU 的选择，会在本书稍后说明。



▲ 计价以 Socket 为准，不是以服务器或内核为准

2. 中大型企业版本的 vSphere 功能

中大型企业使用的 vSphere 功能上不见得比 Essentials 完整，价钱也不见得贵，主要还是将真正需要的功能对应到企业服务上。事实上，中大型企业运用的 vSphere 版本较在乎的是机房的管理和云计算，因此在功能上较偏重服务的可用性（HA）和管理功能。下面就来看看这四个版本的内核部件。

▶ **vSphere 标准版**

- 针对单个内核授权，一个 CPU 插槽需要买一个授权。
- 需要单独购买 vCenter Server。
- 每一个授权 CPU 只能有 6 个内核。
- 单个服务器最多 256GB 内存。

- 使用 ESX 或 ESXi，视状况而定。
- VMware vStorage VMFS（可安装在独立存储或 ESX 的本地硬盘）。
- 四路的虚拟 SMP。
- VMware vCenter Server Agent。
- VMware 的 vStorage API 使用。
- VMware 副本程序（VMware Consolidate Backup，以下简称 VCB）。
- VMware vCenter Update Manager，升级管理员。
- VMware HA 高可用性。
- VMware 存储 Thin Provosioning。
- SNS 需单独购买，至少一年。
- 每一个 CPU 插槽售价 795 美元（此为引用售价，以当地具体售价为准）。

▶ **vSphere 进阶版**

- 标准版的所有功能。
- 每一个 CPU 插槽允许有 12 个内核。
- VMware Vmotion。
- VMware Hot Add。
- VMware Fault Tolerance。
- VMware Data Recovery。
- VMware vShield Zones。
- 每一个 CPU 插槽售价 2 245 美元（此为引用售价，以当地具体售价为准）。

▶ **vSphere 企业版**

- 进阶版的所有功能。
- 每一个 CPU 插槽允许有 6 个内核。
- VMware Storage Vmotion。
- VMware DRS。
- VMware DPM。
- 每一个 CPU 插槽售价 2 875 美元（此为引用售价，以当地具体售价为准）。

▶ **vSphere 企业加强版**

- 企业版的所有功能。
- 每一个 CPU 插槽允许有 12 个内核。
- 无内存制约。
- VMware vNetwork Distributed Switch。
- VMware Host Profiles。
- 每一个 CPU 插槽售价 3 495 美元（此为引用售价，以当地具体售价为准）。
- 自带 Cisco Nexus 1000v 的版本为 3 995 美元（此为引用售价，以当地具体售价为准）。

由于中大型企业的 vSphere 版本功能很多，在本书的稍后章节会一一介绍，先不多做说明。

3. 成本计算

vSphere 中大型企业版本必须一一购买部件，而企业也必须先准备第三方服务，如 MS-SQL 2005 Expres 以上就是必须的。此外也必须另外付费购买 vCenter Server，和 Essentials 本身就附有

vCenter Server 是不同的。表 2 是企业购买 vSphere 不同版本时的成本计算。

表 2　vSphere 中大型企业版成本计算表

此为引用售价，以当地具体售价为准	标准版	进阶版	企业版	企业加强版
Package	$795	$2 245	$2 875	$3 495
金牌或白金 SNS	$273/323	$472/561	$604/719	$734/874
vCenter Server 标准版（无限主机）	$4 995	$4 995	$4 995	$4 995
vCenter Server 基本版（三台主机）	$1 495	$1 495	$1 495	$1 495
vCenter SNS 金牌或白金	$545/645	$545/645	$545/645	$545/645
成本	$3 108 至$6 758	$4 757 至$8 446	$5 519 至$9 234	$6 269 至$10 009

> 注意
> **部署 VMware vSphere 的真正成本**
> 使用 vSphere 的初期成本看来甚高，但以企业省下来花费来看，还是很值的，且在短期内就可以回本。简单来说，一台服务器的成本就可以买一套 vSphere，以 30 万左右的服务器来说，上面运行数十个 VM，就可以省下几十台服务器的成本，何乐而不为呢！稍后会有更多成本精算的实例。

2.5 最新版的 vSphere 4.1

在本书即将付梓之前，VMware 隆重推出了 vSphere 4.0 的小改版 4.1，虽然说是小改版，但还是有些新的功能，下面就来看看 vSphere 4.1 的新功能，不过本书大部分的操作还是以 4.0 为主，读者在阅读时要留意一下。

vSphere 4.1 改正了将近 150 个 vSphere 4.0 的问题及功能，并且在许多性能上有重要的加强。

> 注意
> **不会再有 ESX 了**
> vSphere 4.1 最重大的改变，就是将原来主流的 ESX 撤销，并且将以 ESXi 为热点。VMware 宣称在下一个改版中，所有的 Hypervisor 都将会是 ESXi，并且更名为 vSphere Hypervisor。主要的原因就是 ESXi 较轻巧，并且能落实所有 ESX 的功能。

2.5.1　存储方面的升级

vSphere 4.1 在各个方面都有很大的升级，主要是集中在功能、存储、网络等部分，我们就先来看看在存储方面的进步。

1. Performance Reporting

vSphere 4.1 提供了每一台主机、存储设备、存储卡、存储路径的流量及延迟报告，甚至是每一个 VM、Datastore、VMDK 等。如果你在使用 vSphere 4.1 时发现访问硬盘或是 SAN 时变慢，那么就可以从这个报告中找到症结。

2. 存储 I/O 总控

这是大家津津乐道的功能，就是可以总控存储设备的 QoS，确保每一个 VM 在使用存储资源时，都能够使用到足够的带宽。

3. 支持 8GB 的 HBA

网络存储越来越主流，vSphere 4.1 也不落人后，当前 HBA 已经到了 8GB 的速度，因此 vSphere 4.0 时还无法赶上，但在推出新版本之后，也可以使用最新的 8GB HBA 访问网络存储设备了。

4. 磁盘数组集成 API（VAAI）

这是 VMware 和磁盘数组之间新的通信协议界面，让 vSphere 有关存储设备的操作更快捷，如 SVMotion 或是从模板生成新的 VM。当然这必须在数组端就落实集成，因此要有支持 VAAI 的功能，存储设备厂家必须加快脚步集成。

2.5.2 网络部分的集成

网络一直是 vSphere 4.0 最复杂的部分，而 vSphere 4.1 更是加强了网络方面的功能。

1. 网络 QoS

如果你有安装 vDS，那么就可以针对其 I/O 进行总控，包括配置其优先序，流量内容的管理，让用户能更有效管理 vSphere 下的网络资源。

2. 负载平衡 Teaming

从前在 vSphere 4.0 只支持 NIC Teaming，但是在 vSphere 4.1 中，特别针对 NLB 的 Teaming 有更多的功能，主要还是为了加快较复杂 vSphere 环境中的网络速度。

2.5.3 集群的功能加强

在 vSphere 中，最被大家看好的功能就是 HA，而 vSphere 4.1 也针对了 HA 作了加强。

1. VMotion 的加强

在 vSphere 4.1 中，可以在 1GB 的网卡并发进行 4 个 VMotion，在 10GB 的网卡更可以并发进行 10 个 VMotion，这简直是不可思议的速度。而新版的 4.1 也增强了 EVC 的兼容性。大家可以在本书 VMotion 的章节中了解 EVC 和 VMotion 的细节。另外针对 HA，vSphere 4.1 也在系统健全性上有更深入的诊断功能。

2. 更多的硬件支持

随着新硬件不断上市，vSphere 4.1 也在改版中将这一整年的新硬件放入 HCL 中，包括最新出来的 6 内核 Intel CPU 也成功加入 HCL 了。此外在新版的 vSphere 4.1 中，也支持了 USB 的 Passthrough，换句话说，当你在 ESX/ESXi 主机中插入 USB 设备，可以直接被 VM 认出使用。

3. 内存调优

这也是大家等待最久的功能，主要就是针对 VM 的内存在切换至 Swap 文件之前进行压缩。

这个功能不但可以让固化大小内存的Hypervisor能运行更多的VM，更可以加快VM运行的速度。

4. DRS 配置

DRS 的配置比从前更多样化，以期待能应付越来越复杂的企业环境。

5. VDR 功能改进

VDR 是 vSphere 中副本的机制，新版的 vSphere 4.1 也针对 VDR 做了更多的改良。举例来说，可以在每一个集群中安装 10 个 VDR，并且针对 Windows 7/2008 的 VSS 也支持了。现在的 VDR 支持更多的存储设备，包括 DAS、NFS、iSCSI、FC，并且也可以让目的磁盘支持 CIFS 的共享。新版的 VDR 也支持更高效能的磁盘重复删除功能。

2.5.4 vCenter 的改良

vCenter 是 vSphere 的内核，在 vSphere 4.1 中，vCenter 也升级到 4.1，下面就是 vCenter 4.1 的改良。

1. 只支持 X64 的服务器操作系统

这一点要特别注意。原来的 vCenter 是可以安装在 Windows 2003/X86 上，但在升级到 4.1 之后，原来的 vCenter 无法升级，一定得重新安装。虽然这一点让用户觉得很麻烦，但为了支持更多的 vSphere 4.1 功能，重新安装 X64 的服务器操作系统是值得的。

2. 更高的效能

vCenter 4.1 的激活时间比 4.0 缩短很多，但在 VM 及主机数量少的环境中并没有太明显的改善。

3. 更多的主机及 VM 支持

新版的 vCenter 4.1 可支持 1000 台主机和 10000 个 VM。而每一个 DRS 集群也支持 32 台主机和 3000 个 VM。

结 语

在这一章中，我们理解了 VMware 这家公司及产品的历史，也清楚了 vSphere 的产品特色，对企业主来说，最关注的版本及售价，更在这一章进行了详细说明，在真正准备采购之前，可以掌握预算的分配。

第3章

全面理解vSphere的硬件

关键词：

- 采购或配置任何硬件之前，先到 HCL 遍历兼容性
- 理解 CPU 和 vSphere 的兼容性及功能
- 理解芯片组/主板的兼容性及功能
- 理解内存的数量、种类、速度及插槽数，注意芯片组支持的上限
- 理解 PCI-E 插槽的信道及数量
- 理解存储界面及之间的差异
- 理解硬盘本身的差异
- 理解什么是 RAID
- 理解安装 ESXi 的内置设备
- 理解个人计算机上安装 ESX/ESXi 的热点

大部分要上 VMware 的企业是不太可能全部重新来过的，常见的情况是将目前的硬件重新分配再度使用。因此 IT 人员在上虚拟化之前，要重新查看企业内部的硬设备，因此这些设备是否能在 vSphere 下使用，是 IT 人员评估现有设备的热点。当然如果需要添加新的设备，也必须能使用。

3.1 使用 vSphere 的硬件兼容列表来检查

> **注意** 在进行本章任何硬件的购买或组态前，如果你不确定硬件是否能使用 vSphere，一定要到下面的网址来查看 http://www.vmware.com/resources/compatibility/，该网站的使用方法在本小节有演示，务必先使用这网站，切记！



▲ 任何硬件先来这边检查兼容性

上虚拟化需要哪些硬件？我现有的硬件能上虚拟化吗？这是想上虚拟化的用户最常见的问题。上虚拟化的硬设备并不特殊，当前大部分的硬件产品都能使用在虚拟化上，一般来说，最常出现问题的都在网卡和存储设备上。

> **注意** 此外 vSphere 支持的功能相当多，其中有许多是需要 CPU 或主板支持的，因此在选择服务器时，要特别注意 CPU 和 BIOS 的对应功能，我们在这一章中，就先来认识能在 vSphere 下运行各项功能的硬件选项。

3.1.1 安装 vSphere 的硬件架构

vSphere 需要硬设备和一般企业用的相差无几，以一个完整的 vSphere 环境来说，最主要的是服务器、界面卡和外设如表 1 所示。

表 1　vSphere 的硬设备以及注意事项

种 类	部 件	硬件不兼容时出现的问题
服务器		
	CPU	无法运行 vSphere 中的功能，如 VMotion
	主板（芯片组）	如无法支持够多的内存
	内存	无法发挥 CPU 的性能
	内置网卡	无法安装及使用 ESX 服务器
	内置硬盘总控器	无法安装及使用 ESX 服务器
	内置硬盘	无法安装及使用 ESX 服务器
界面卡（I/O 设备）		
	网卡	无法安装及使用 ESX 服务器
	存储界面下（HBA）	无法安装及使用 ESX 服务器
	远程总控卡	可从网络总控计算机的 KVM（键碟、显示器，鼠标），甚至是 BIOS 都可更改
外 设		
	外部存储	无法安装及使用 ESX 服务器 无法运行 vSphere 中的功能，如 VMotion
	网络设备	无法运行 vSphere 中的功能，如 VMotion

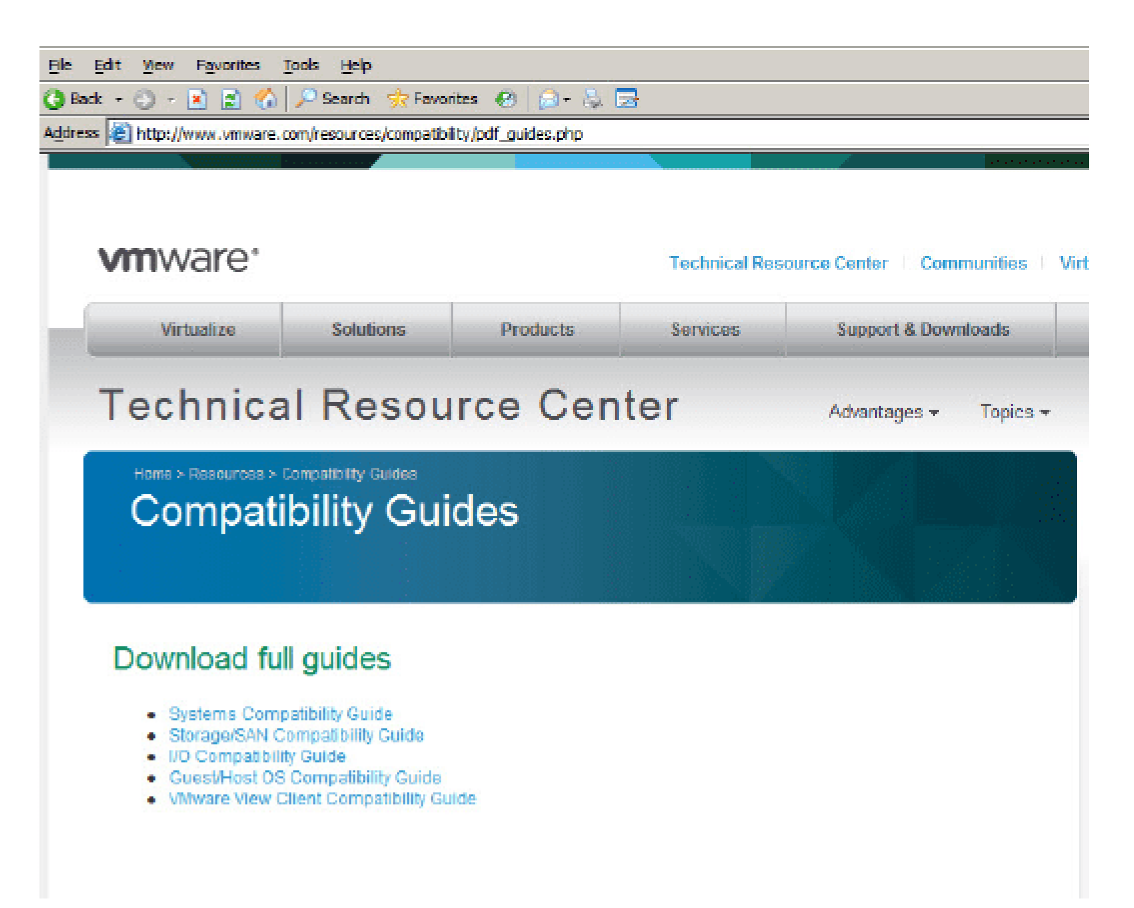
3.1.2　使用 HCL 来遍历兼容性

第一次上虚拟化的 IT 人员最常产生的疑问，就是“我当前的硬设备能使用 vSphere 吗？”由于 vSphere 并不是一般的桌面型 Windows 操作系统，因此支持的硬件通常以服务器环境为主。由于服务器环境需要的硬件较严谨，因此 vSphere 所支持的硬件并不如 Windows 系列。但当前大部分的服务器厂家在开发时，也都考虑到买家很可能使用虚拟化，因此也多朝兼容的方向发展。

为了帮助用户在购买或使用现有硬件时能确定是否能使用 vSphere，VMware 也提供了一个网页功能，不但可以评估现有硬件，也能在增购新硬设备之前确定所有的功能都可以使用。先来看看这个硬件兼容性列表（Hardware Compatibility List，以下简称 HCL）。

1. 可下载的 HCL 文档

一般来说，可以在 http://www.vmware.com/resources/compatibility/pdf_guides.php 下载完整的 HCL 文档，包括服务器、CPU、存储及 IO 等。PDF 文档较为详细，有完整的列表可查询，但缺点是无法找到较新的硬设备，较适合新选购设备时引用。VMware 会不定期补丁这些文档，读者们可以在需要时上网下载最新版的 PDF 文档。

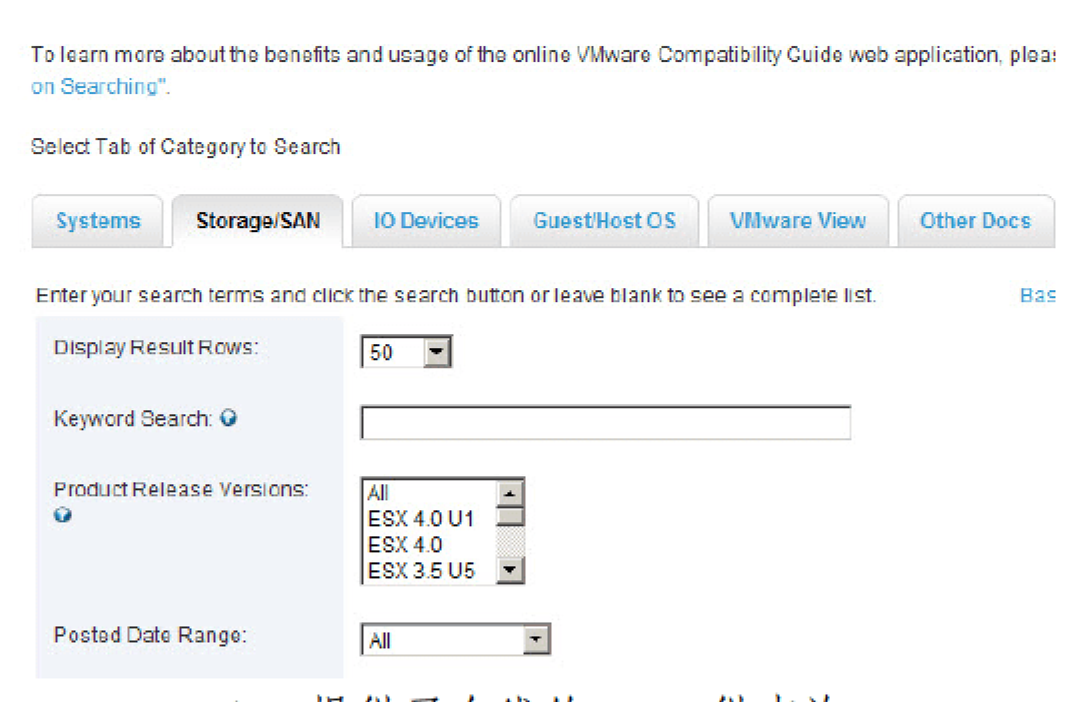


▲ 可以在此下载完整的 PDF 文件，VMware 也会随时修订补丁

2. 使用在线 HCL 遍历服务器硬件兼容性

VMware 提供了一个在线 HCL 服务，可以遍历服务器、存储设备和 IO 界面卡。但无较完整的列表可供引用，如 CPU 兼容性列表，较适合于现有设备的查询。

可以在http://www.vmware.com/resources/compatibility/这个网址进入兼容性遍历。在这个网页中，可以看到 vSphere 有 6 个选项卡，但需要我们注意的是系统（Systems）、存储（Storage/SAN）以及 I/O 设备（通常是网卡和存储卡）。下面以一个具体的示例来看看 HCL 的使用方法。



▲ 提供了在线的 HCL 供查询

▶ 用 HCL 来遍历硬件兼容性

1. 如果我们现在有一台 DELL PowerEdge 2950，但不知是否支持 vSphere 的各项功能，我们可以用 HCL 来遍历。首先进入网页，并且在 Keyword Search 的条形框键入 PowerEdge 2950。

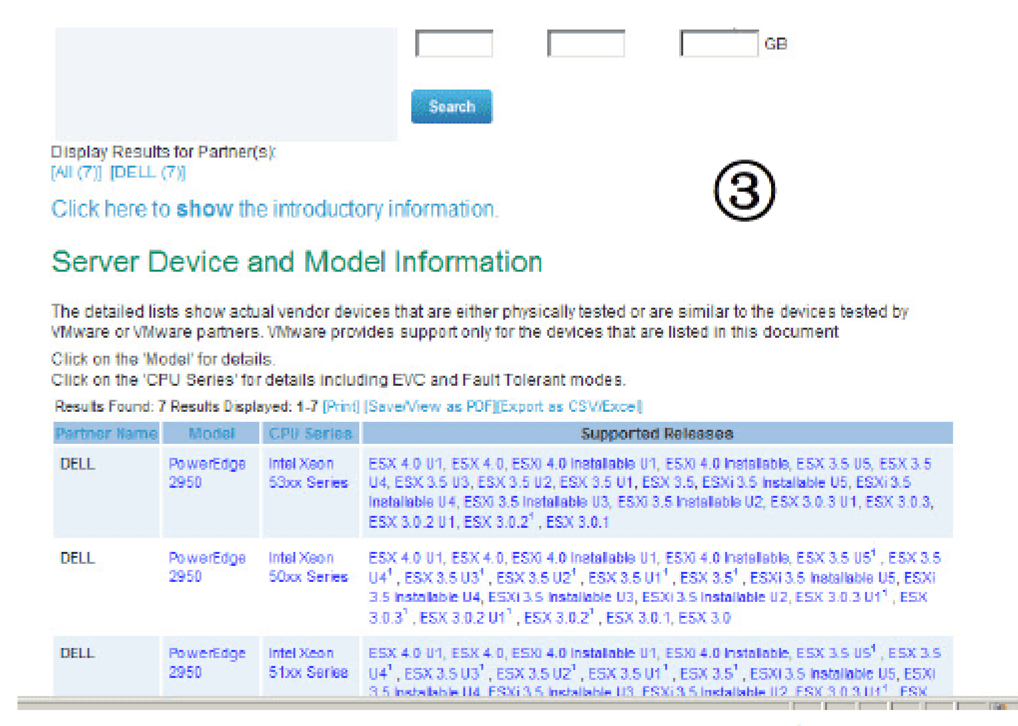
2. 然后可以直接到页面最下方单击 Search，但也可以针对更细致的选项来检视，如针对 ESX 的版本、制造商名称、机壳大小以及 CPU 型号等。由于 ESX 的功能很多需要依托 CPU 落实，如 EVC（在本书稍后会提到）或 Fault Tolerant，也可以检视看看是否有支持这些功能。



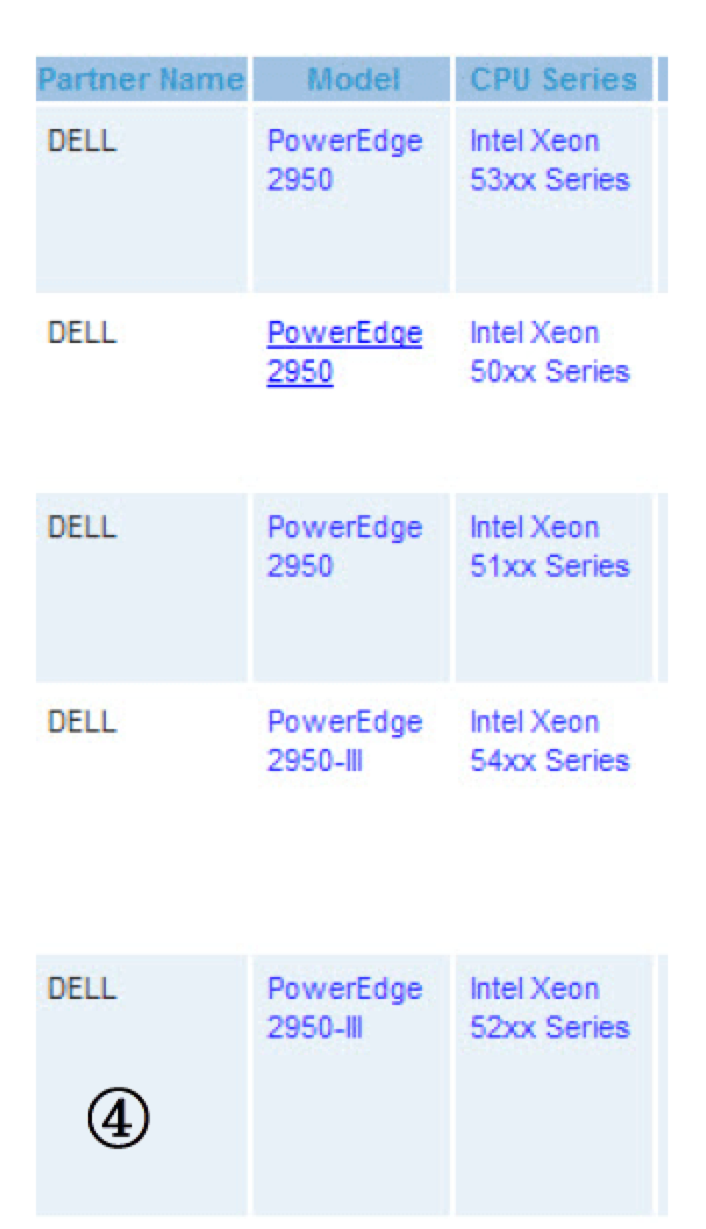
▲ 键入要遍历的硬件　　　▲ 可以直接查询，也可以自定义查询的条件

3. 在单击 Search 按钮之后，会给出匹配条件的产品。可以看到机型、CPU 系列以及支持 ESX/ESXi 的版本别。

4. 大部分的服务器厂家在将产品更新换代时，还会沿用原来的产品型号，因此可以进行机型的详细规格查询。当我们按下机型选项卡时，会弹出如右图所示的窗口，会显示型号、厂牌、CPU 种类、机壳大小、CPU 插槽数以及最大内存数量。



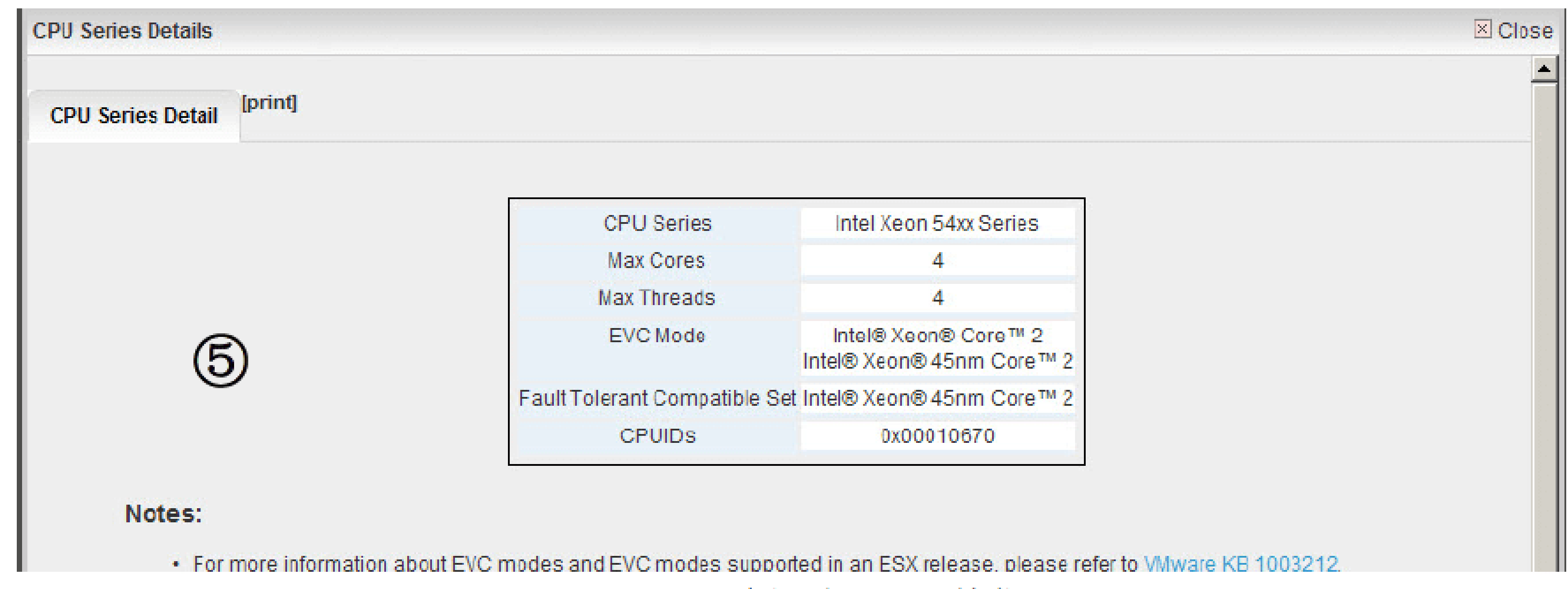
▲ 会给出匹配的产品，后面的链接为支持的 VMware 产品类别



Partner Name	Model	CPU Series
DELL	PowerEdge 2950	Intel Xeon 53xx Series
DELL	PowerEdge 2950	Intel Xeon 50xx Series
DELL	PowerEdge 2950	Intel Xeon 51xx Series
DELL	PowerEdge 2950-III	Intel Xeon 54xx Series
DELL	PowerEdge 2950-III	Intel Xeon 52xx Series

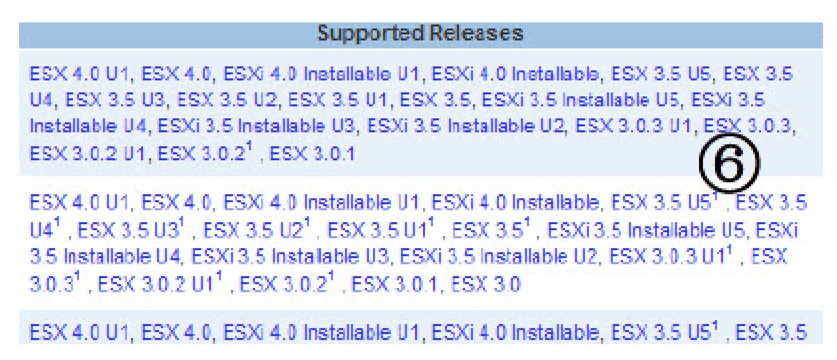
▲ 如 Dell PowerEdge 2950 从 2002 年就有此型号，但规格一直在进步

5. 当我们按下 CPU 型号选项卡时，会弹出如下图所示的窗口，会显示 CPU 系列、内核数、进程数以及功能（如 EVC 或 FT），也会有详细的 CPU Stepping 编号。

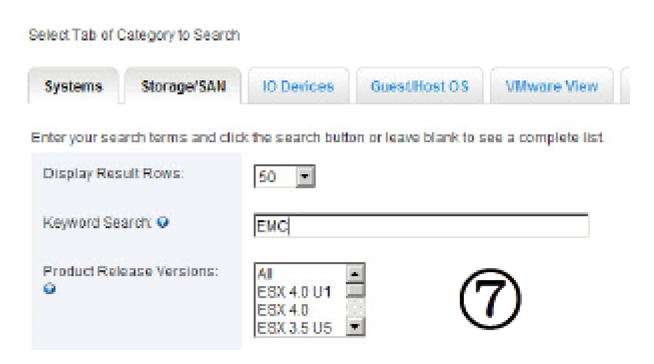


▲ 显示详细的 CPU 数据

6. 选择哪个 ESX 版本，就会弹出这个 ESX 的支持以及 BIOS 的支持。
7. 选择 Storage/SAN 选项卡，并且键入关键词，一样也可以选择厂家、ESX 版本等。最主要的还是选择存储设备的类型。如 SAN、iSCSI、NAS、SAS 等。

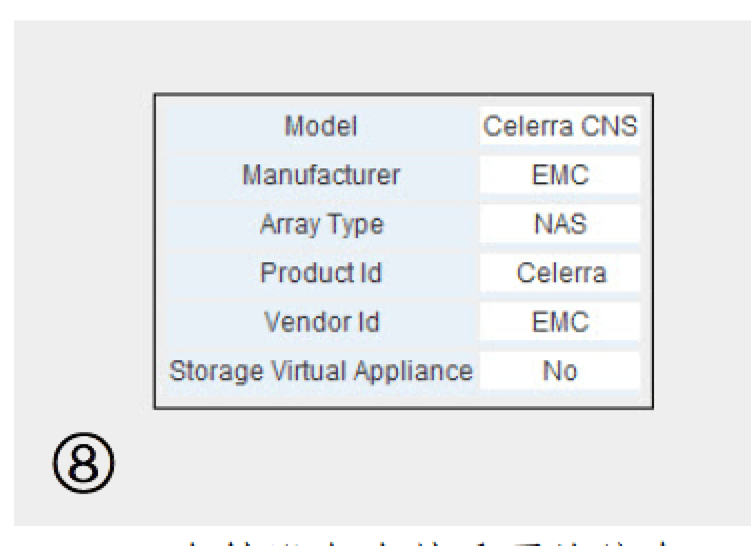


▲ 给出对不同 ESX 版本的支持



▲ 选择存储设备产品的演示

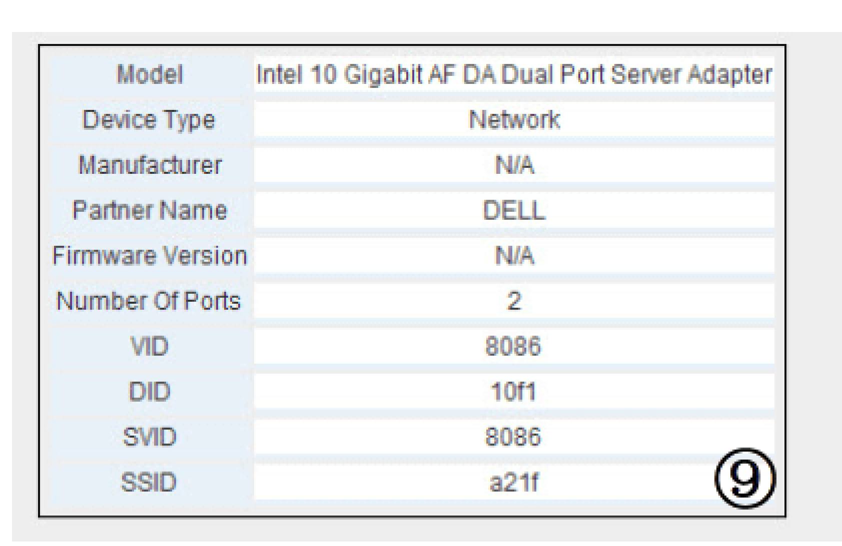
8. 当找到匹配的结果时，可以查看到厂家、型号等。选择该机型，便会弹出厂家、型号以及界面等信息。
9. 选择 I/O 设备，并且键入关键词，如 Intel Pro/1000。并且开始选择。也会找到对应的设备，单击该设备的详细数据，会弹出名称、厂家、类型等。如果是多个通信端口的界面卡，也会弹出卡的通信端口数量。



Model	Celerra CNS
Manufacturer	EMC
Array Type	NAS
Product Id	Celerra
Vendor Id	EMC
Storage Virtual Appliance	No

⑧

▲ 存储设备有其重要的信息



Model	Intel 10 Gigabit AF DA Dual Port Server Adapter
Device Type	Network
Manufacturer	N/A
Partner Name	DELL
Firmware Version	N/A
Number Of Ports	2
VID	8086
DID	10f1
SVID	8086
SSID	a21f

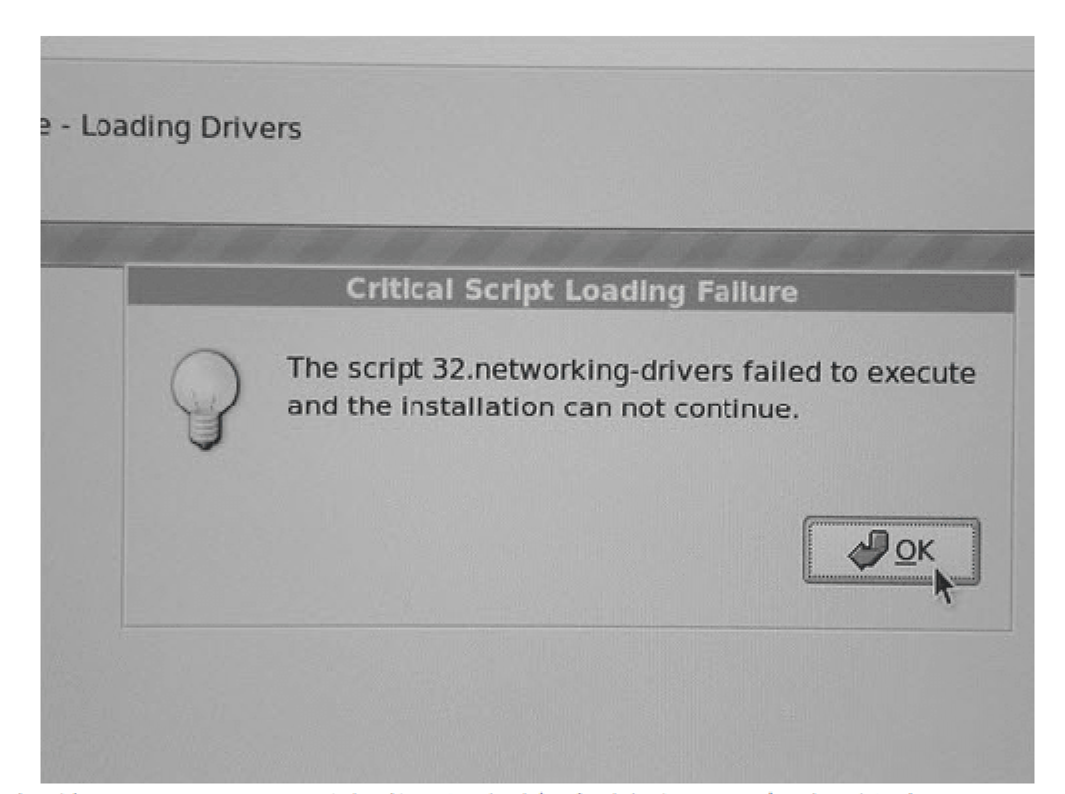
⑨

▲ I/O 设备也有其对应的产品及信息

注意

最常弹出问题的设备

大部分在使用 vSphere 时，最容易发生兼容性问题的地方就是 I/O 设备，通常就是网卡和存储界面卡。一般来说，PC 产品的界面卡在 ESX 上是无法安装的，因此在选购时，还是以服务器产品为主。



▲ 安装 ESX/ESXi 最常见的兼容性问题多出现在 I/O 设备上

3.2 理解 vSphere 使用的服务器架构

大多数的企业在选购服务器或是使用原有服务器来虚拟化时，常常不知硬设备是否完全和 vSphere 兼容。由于大多数企业购买服务器产品都是直接购买品牌机，当前服务器主流品牌包括 Dell、IBM、HP 等，在购买时，最重要的就是确定要和 vSphere 的功能兼容。

3.2.1 选择服务器的要诀

大部分的品牌服务器都可以用在 vSphere 上，但功能上会有较多的差别。一般来说，服务器使用的都是专用的主板和芯片组， 因此在选择时，也是以芯片组来决定的。一般在上虚拟化时，会以服务器的型号来对应到不同的芯片组上。下面就是服务器的选购原则。

1. 尺寸——塔式、机架式或刀片

一般服务器按照体积分成塔式、机架和刀片三种。如果没有空间的制约，用塔式服务器即可，但要注意兼容性。机架服务器通常兼容性较高，而刀片服务器则是扩展性较强，可以在 CPU 或内存不够时随时扩展。



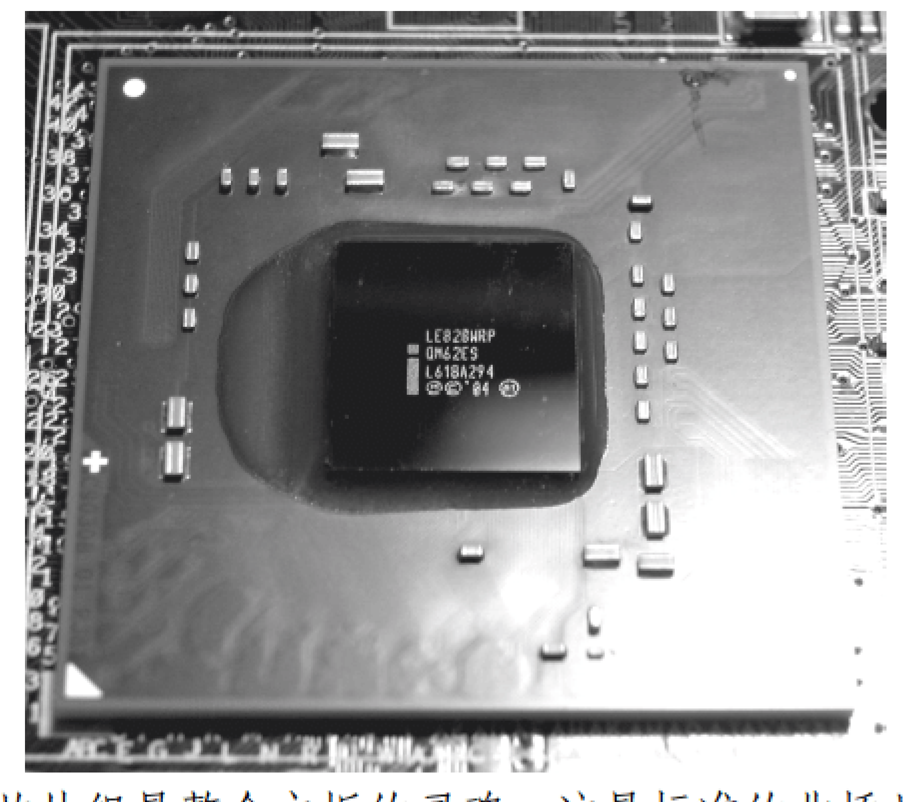
▲ 机房设备使用机架式服务器



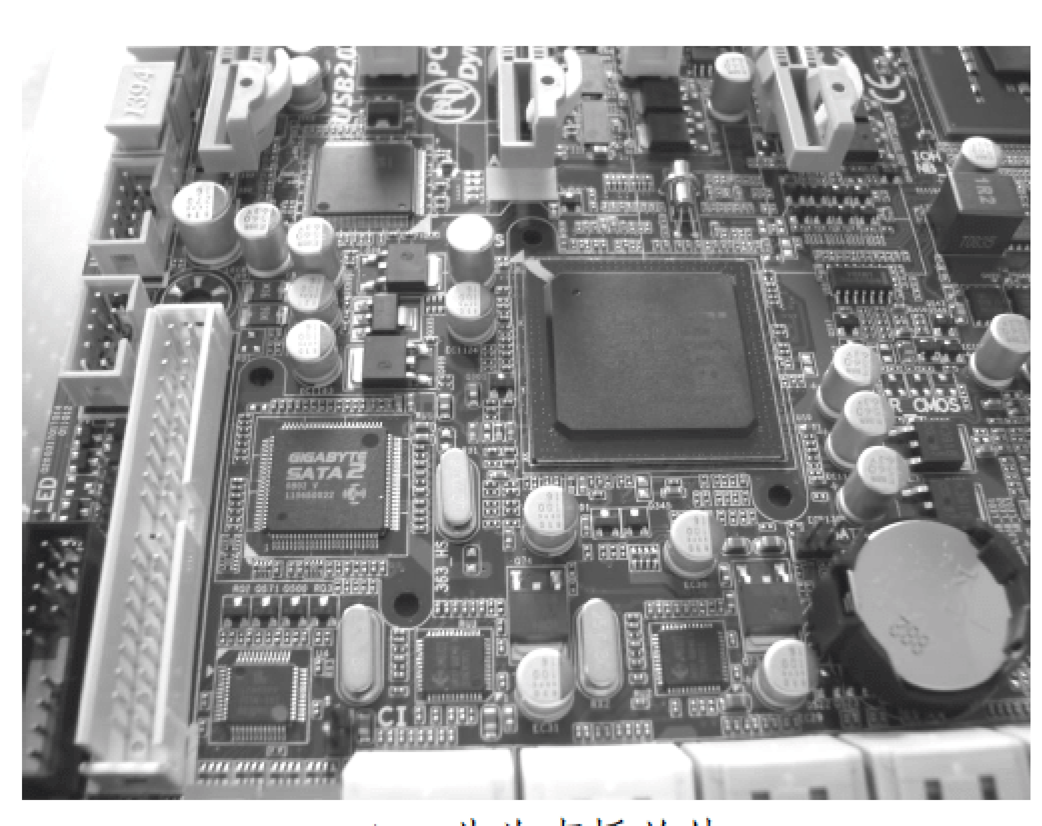
▲ 如果想要有更高的扩展性，使用刀片是很好的选择

2. 功能——芯片组（北桥及南桥）

一般芯片组分为北桥和南桥两部分，北桥负责内存 PCIE、CPU 等快捷设备，南桥则负责 SATA、USB 等慢速设备。南北桥决定了芯片组能支持什么样的硬件。包括内存信道数（内存的速度）、内存的数量、CPU 的种类，以及外置卡的界面种类等。



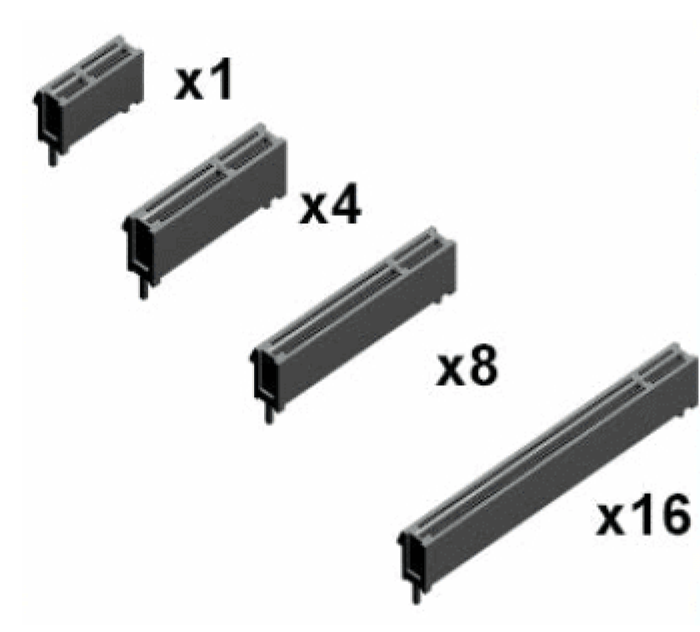
▲ 芯片组是整个主板的灵魂，这是标准的北桥芯片



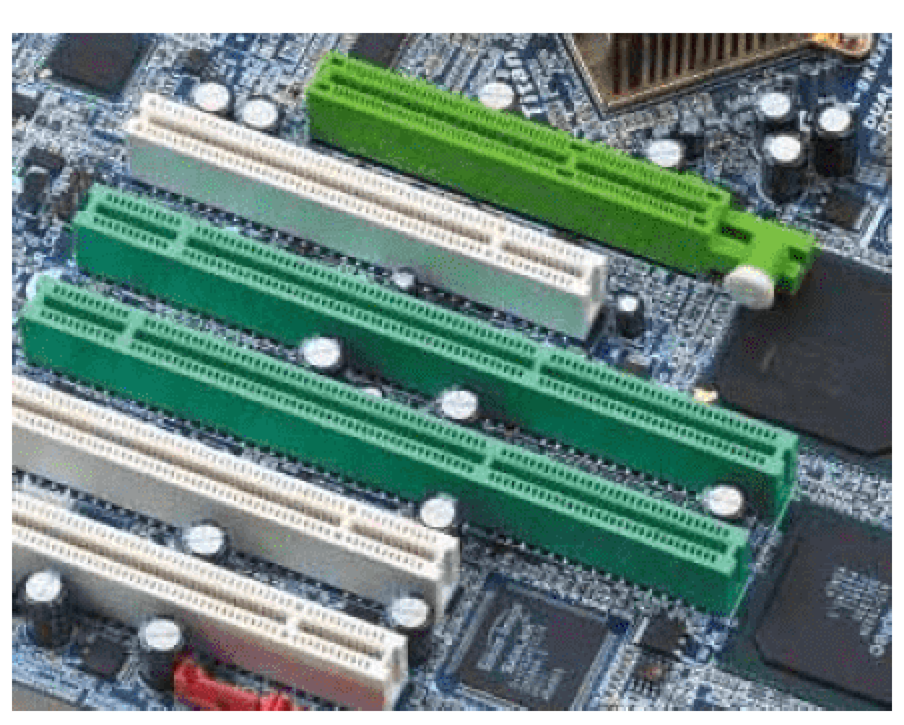
▲ 此为南桥芯片

3. IO 速度——外置界面卡（PCI、PCI-E、PCI-X）

当前界面卡主要为高速的 PCI-E、传统的 PCI 以及带宽较宽的 PCI-X。由于 PCI-E 可以取代 PCI-X，并且有多种不同的速度组合，因此大部分在 vSphere 上安装的外置卡都已经有 PCI-E 的界面，大部分为网卡以及存储设备卡。PCI-X 是和 PCI 兼容的，虽然多出一块界面，但还是可以插在 PCI 上使用。



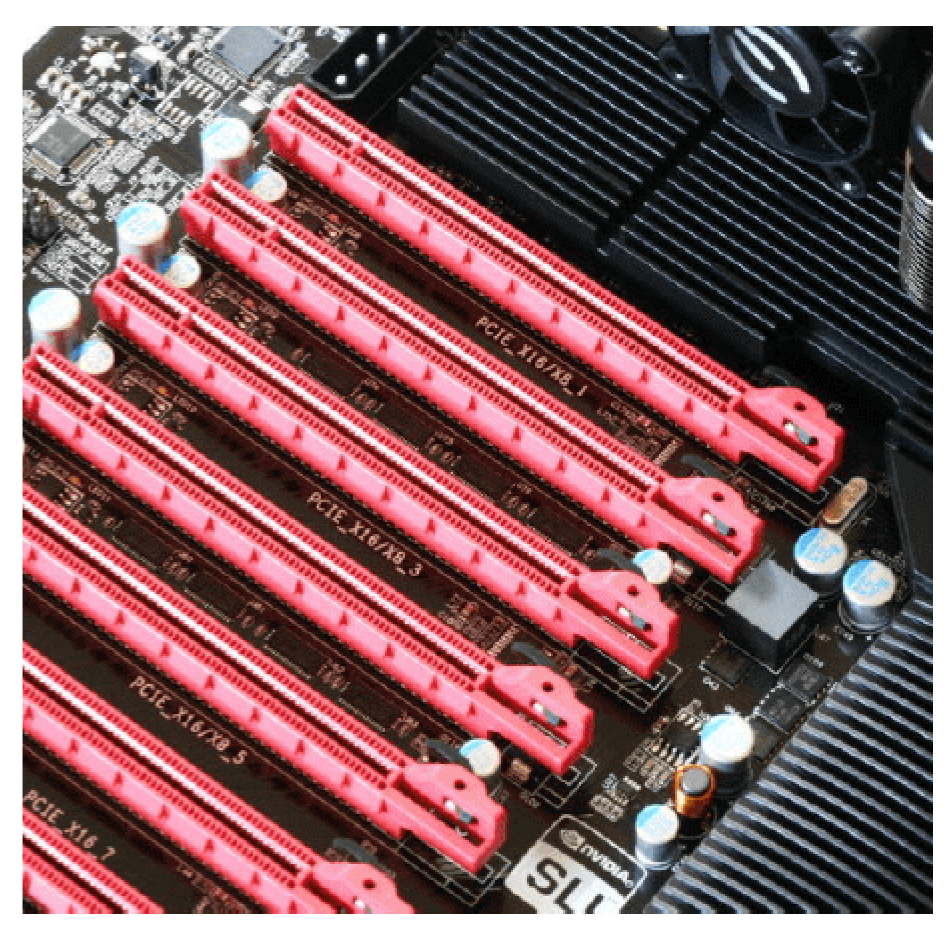
▲ PCI-E 是当前流行的界面，有不同的速度组合



▲ 服务器主板上还是能看到 PCI-X 的界面

4. IO 扩展性——PCI 信道数以及插槽数

一个主板上拥有的 PCI 信道数是固化的。由于新的 PCI-E 利用了多信道，因此一片界面卡可能会占用多个 PCI 信道，从而制约了 PCI 卡的数量。另外制约 PCI 外置卡数量的关键就是 PCI 插槽数。也许读者觉得这不重要，但当你的 vSphere 需要多个物理网卡来做硬件对应时，越多 PCI 插槽和 PCI 信道数量，可以决定一台物理服务器上 VM 的真正数量。

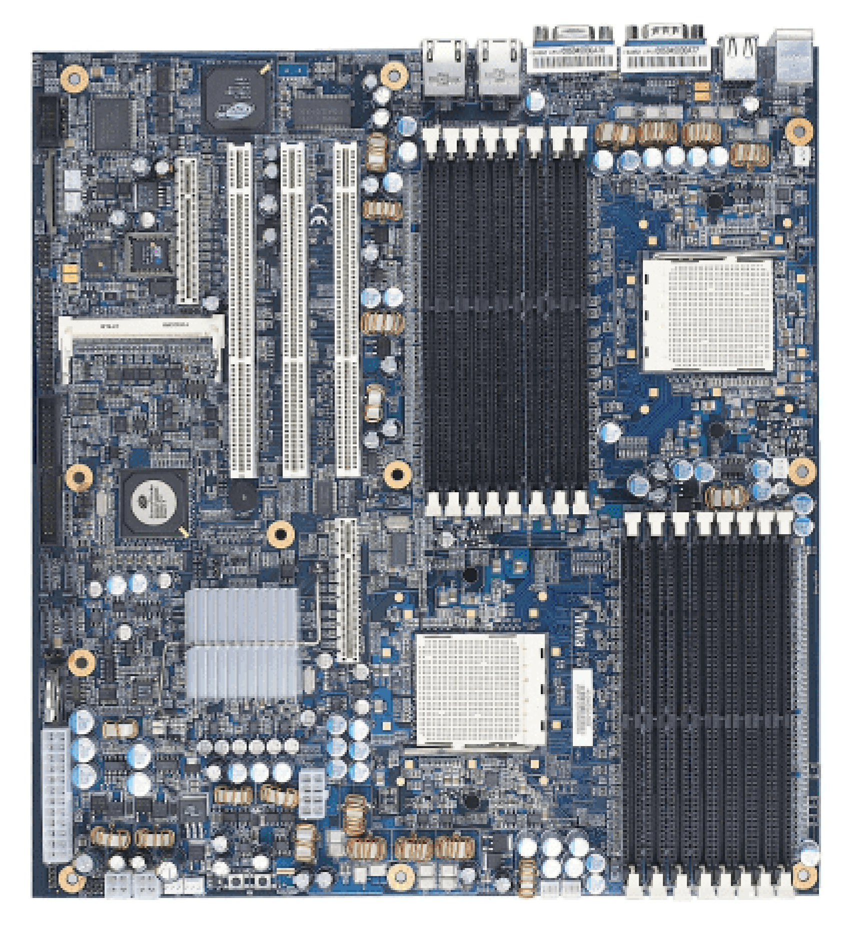


▲ 一片好的主板上应该是有非常多的 PCI-E 插槽

5. 内存——主板内存规格的选择

一台服务器上内存总数量决定于芯片组，而 vSphere 也有对内存做制约。一般服务器的内存制约有 64GB、128GB、192GB 和 256GB 等，但真正会制约内存的还是内存的插槽数。一般来说，我们会选择最多插槽的主板以应付日后的需要。

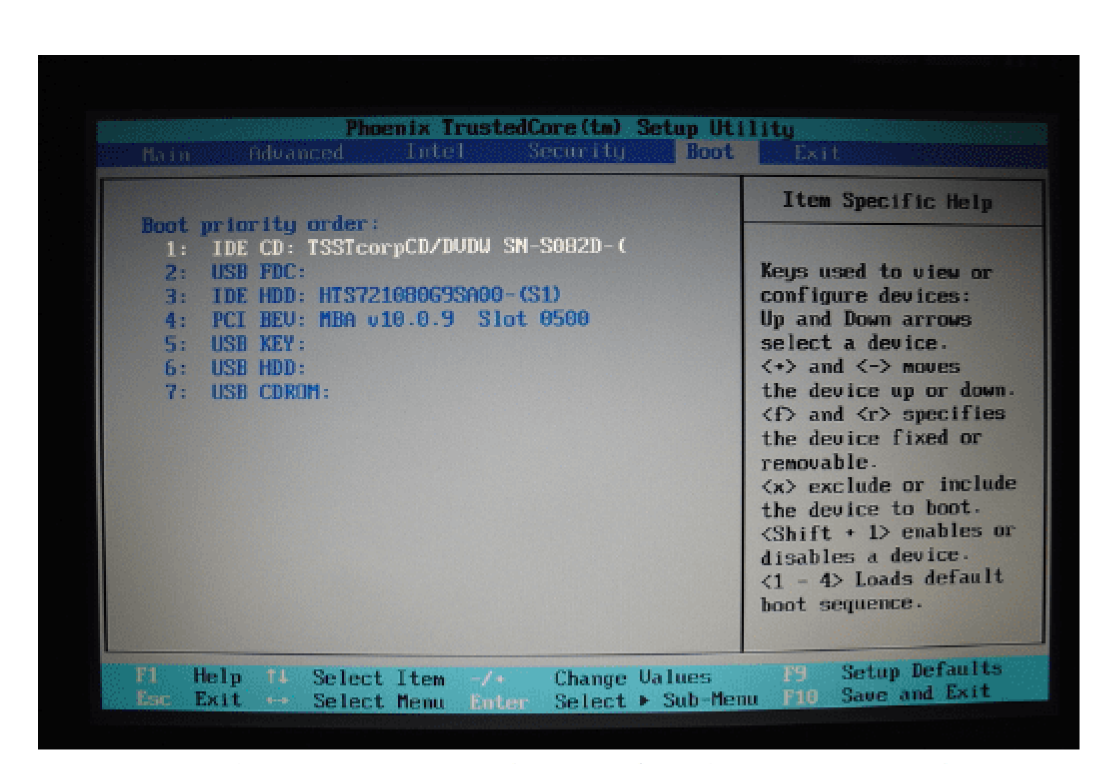
另外就是内存的信道数。这意味着 CPU 一次能处理几个插槽的内存，当前有单信道、双信道和三信道三种，信道数越多的内存效能越好。另外就是内存的速度，这和计算机的时钟有关，这个数字当前有 1066MHz、1333MHz 及 1600MHz 等，当然是数字越高速度越快。



▲ 除了信道数、产品种类和速度外，主板上插槽数越多越好

6. 引导弹性——引导设备种类

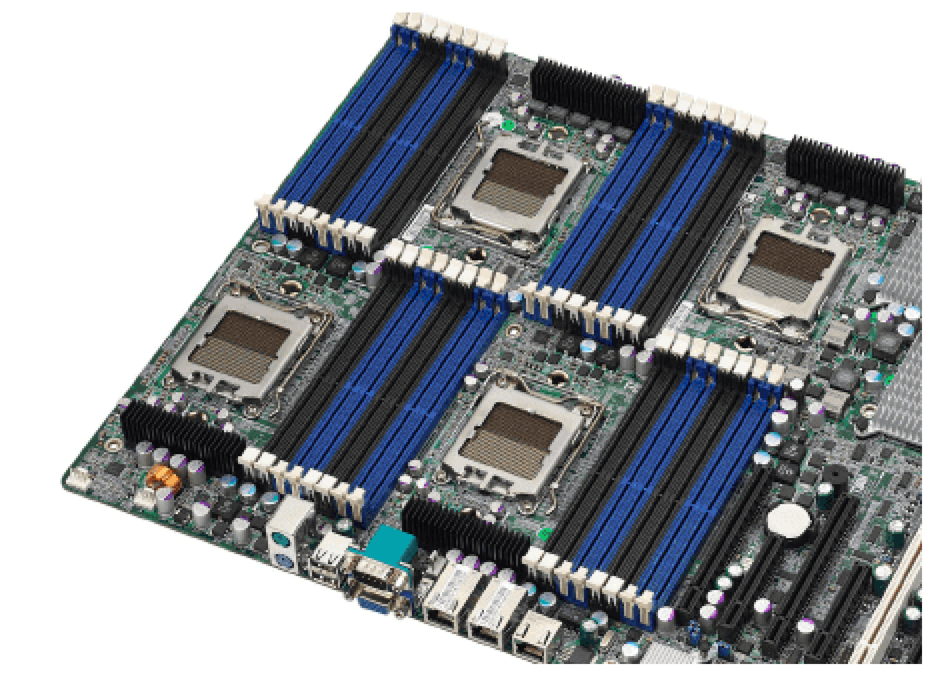
主板引导设备的种类和顺序一定要够完整，除了标准的光盘、硬盘之外，一定要有不同的 USB 设备引导以及网卡 PXE 引导。举例来说，vSphere 中的 ESXi USB 可以通过 USB 引导进入 Hypervisor，而许多 Dell 和 IBM 的服务器更有专门的 Embedded Hypervisor 插槽激活内置的 Hypervisor。



▲ 引导选项可以决定主机引导方式的弹性

7. CPU 扩展性——CPU 插槽数

主板上能插上的物理 CPU 个数通常有 1、2、4 个，如果每一个 CPU 又有多内核（如 8 内核），那么一个主板上最多可以有 32 个 CPU 内核，这样就能更平顺地运行多个 VM 了。



▲ CPU 插槽数至少要有两个，四个的话速度已经飞快了

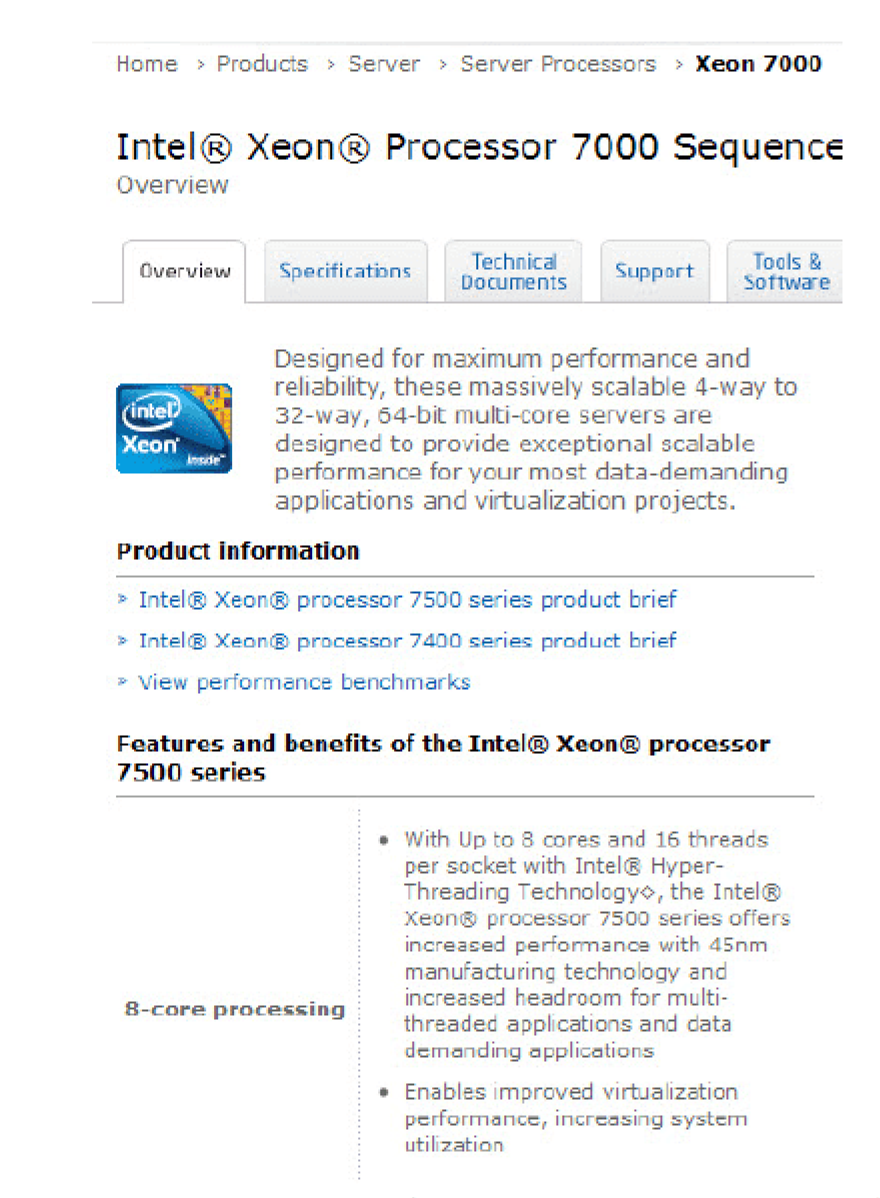
3.3 理解 vSphere 使用的 CPU

虚拟化在这两年成为显学，硬件上最功不可没的就是 CPU。从 2005 年开始，Intel 和 AMD 就开始在 CPU 置入提速或实现虚拟化的命令集，再加上多内核及高时钟让一个 CPU 上能运行上百个操作系统。但一个 CPU 能使用的时间很长，通常旧计算机的 CPU 不见得能兼容于所有 vSphere 的功能，因此我们就在这一小节来看看 CPU 的分类及汇总。

3.3.1 理解 vSphere 支持的 CPU

vSphere 支持大部分的服务器专用 CPU，如 Intel 的 Xeon 系列或是 AMD 的 Opteron 系列，其他的差别就是时钟、内核数等。

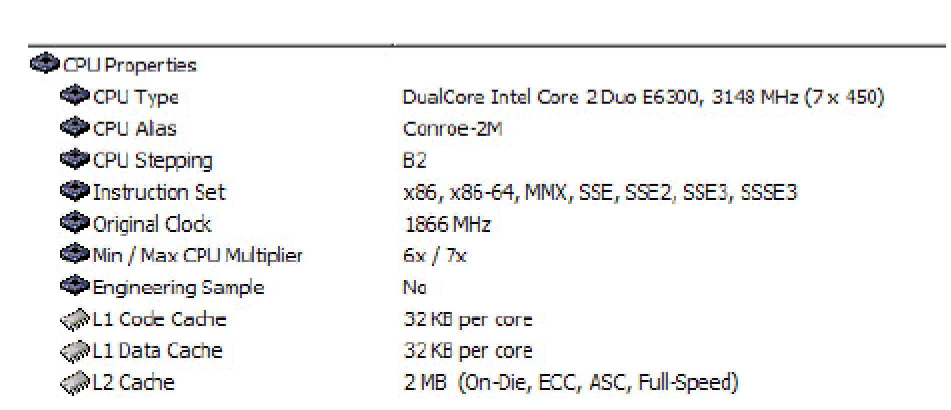
为了 VM 的速度，我们当然推荐高时钟多内核的 CPU，但最高时钟的 CPU 通常是最贵的。笔者的建议是，购买最高时钟下一级的产品。举例来说，Xeon 系列当前最高级的是 7500 系列的 7560，那么购买时就选择下一级的 7550，虽然性能只有最高级 7560 的 80%，但成本可以只不到一半，是最好的选择。



▲ Intel 的 CPU 功能强大，是最让人省事的产品

1. 64 比特

由于 vSphere 是 64 比特的 Hypervisor，非 64 比特的 CPU 是完全不能使用的，这一点在购买 CPU 时一定要问清楚。



▲ 当前大部分的 CPU 都是 64 比特的，买错的概率不高

2. 支持完整的 vSphere 命令

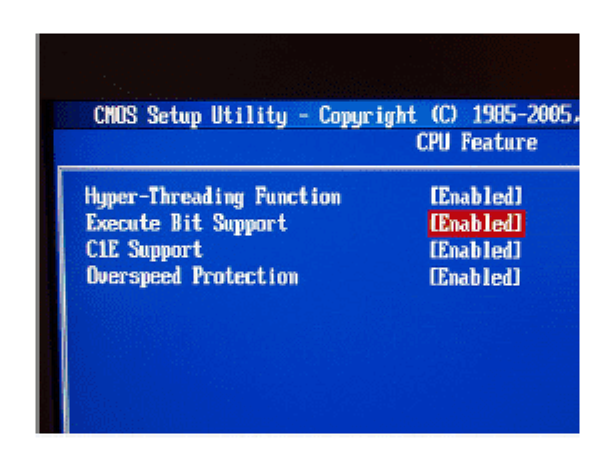
CPU 要支持特殊的命令才能完整使用 vSphere 全部的功能，如在做 VMotion 时，需要 SSE4 才能完整使用 EVC 功能（请参阅 VMotion 章节）。在购买时，一定要和工程师确定这些功能。

CPU Series:
Intel Xeon 74xx Series
Intel Xeon 75xx Series
Intel Xeon Dual Core (Sossaman) Se
Intel Xeon MP (Cranford/Potomac) Se

▲ 可以在 HCL 上用 CPU 来找是否支持完整的 vSphere 命令

3. Intel 的 VT-X 或 AMD 的 AMD-V

这是 CPU 中专门运行虚拟化的命令，没有的话是无法运行的。当前大部分的服务器 CPU 都具有此功能，用户只要稍注意就行。

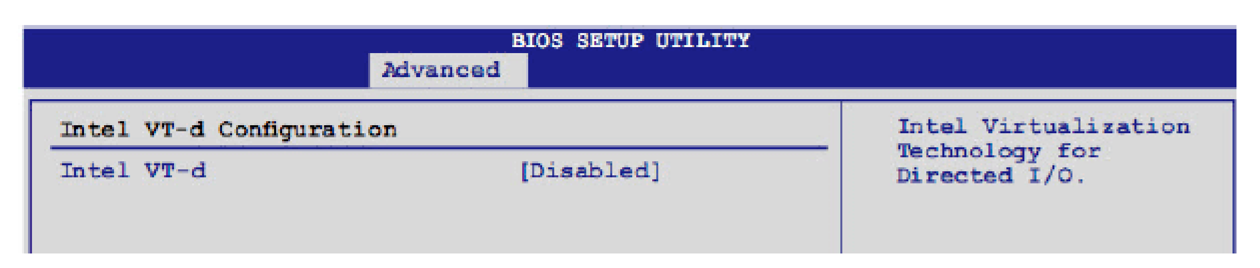


▲ 在 BIOS 中可能是以 Execute Bit 的方式来打开

4. VT-D 功能

许多人在挑选 CPU 时，只注意 VT-X，但要真正发挥 vSphere 的所有功能，还有一个就是 VT-D。VT-D 可以让 vSphere 上的 VM 直接使用硬件，通常是网卡或是存储界面卡，因此 VT-D 的功能也

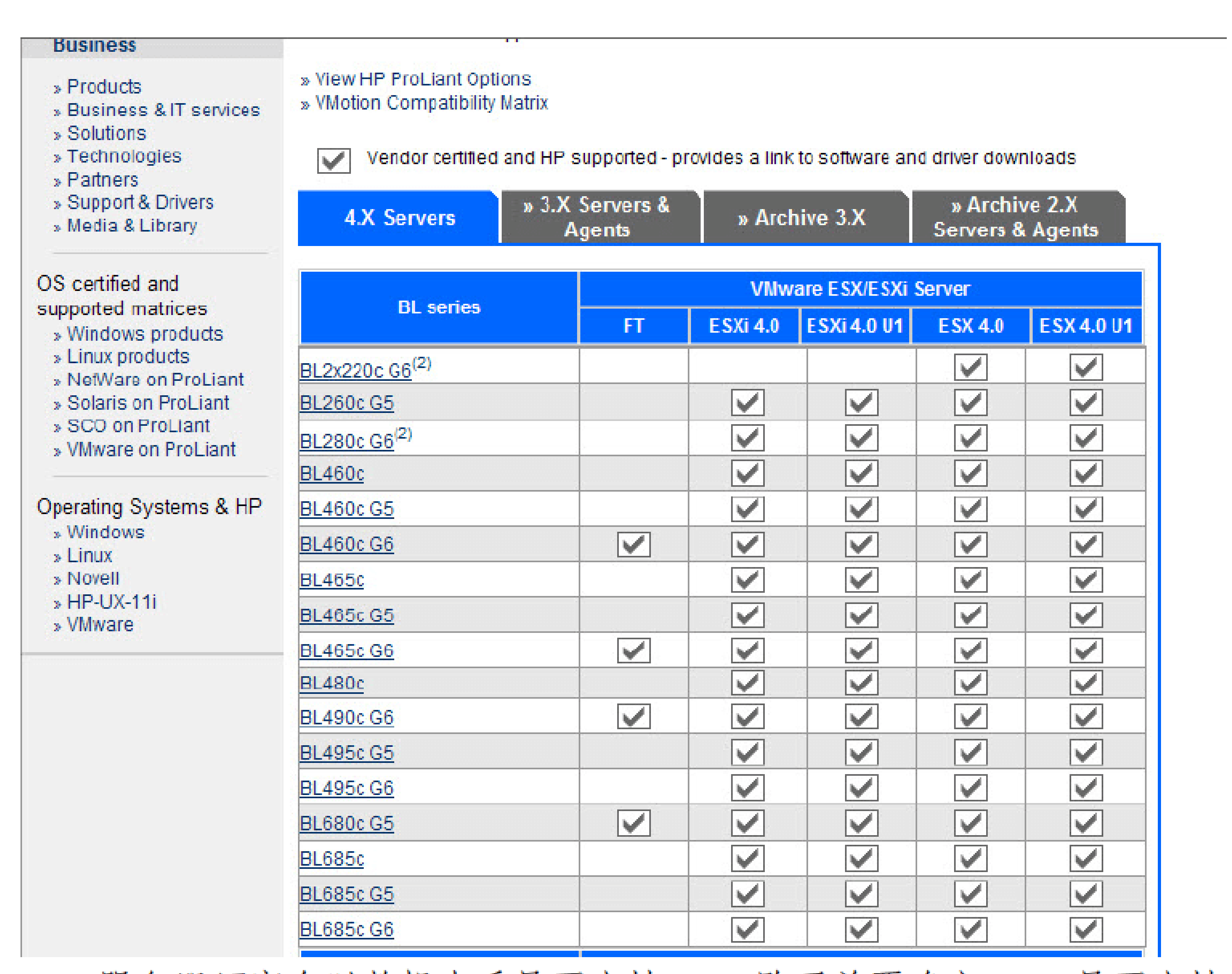
很重要。特别要注意的是，大部分的服务器 CPU 都有这个功能，但主板的 BIOS 中一定要有这个功能才能打开 VT-D，在购买服务器或主板之前，一定要先弄清楚。



▲ 主板一定要有打开 VT-D 的功能，在购买前一定要询问

5. Fault Tolerance 功能

冗余（Fault Tolerance，以下简称 FT）功能在 vSphere 中是新加入的。这个功能是利用 CPU 的命令来落实双机待命的实现。CPU 必须支持这个功能才行。本书稍后会有完整的篇幅介绍 FT 的实践。



BL series	VMware ESX/ESXi Server				
	FT	ESXi 4.0	ESXi 4.0 U1	ESX 4.0	ESX 4.0 U1
BL2x220c G6(2)				✓	✓
BL260c G5		✓	✓	✓	✓
BL280c G6(2)		✓	✓	✓	✓
BL460c		✓	✓	✓	✓
BL460c G5		✓	✓	✓	✓
BL460c G6	✓	✓	✓	✓	✓
BL465c		✓	✓	✓	✓
BL465c G5		✓	✓	✓	✓
BL465c G6	✓	✓	✓	✓	✓
BL480c		✓	✓	✓	✓
BL490c G6	✓	✓	✓	✓	✓
BL495c G5		✓	✓	✓	✓
BL495c G6		✓	✓	✓	✓
BL680c G5	✓	✓	✓	✓	✓
BL685c		✓	✓	✓	✓
BL685c G5		✓	✓	✓	✓
BL685c G6		✓	✓	✓	✓

▲ 服务器厂家会以整机来看是否支持 FT，购买前要确定 CPU 是否支持

3.4 网络设备

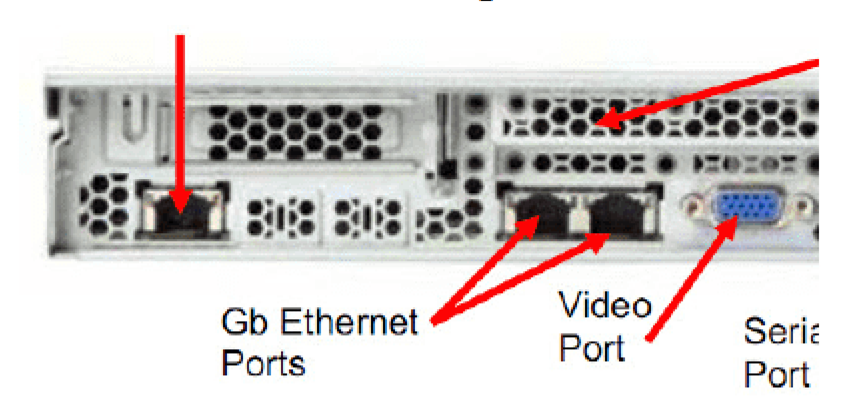
网络设备分成网卡以及外置网络设备。一般来说，服务器主板上都会有内置的网卡，但网卡的速度、功能、厂牌型号等都会影响 vSphere 的使用及功能。在外设上面，最常遇到的兼容性问题就是交换机或路由器。由于 vSphere 支持虚拟路由器（Cisco Nexus 1000V），因此在分组交换会有一些注意事项。

3.4.1 网卡的注意事项

所有的服务器都有内置网卡，网卡是决定 vSphere 效能及功能的最重要因素之一，我们就来看看选购或配置时的注意事项。

1. 内置网卡的品牌

在兼容性上，还是一句老话，使用前不要忘了到 HCL 网站检视，但当前来说，对 vSphere 最支持的厂牌还是 Intel 和 Boradcomm 两个，如果你在 HCL 中找不到服务器的牌子，可以询问该服务器的网卡芯片组是否为以上两个品牌，能在 vSphere 下常规运作的概率较高。



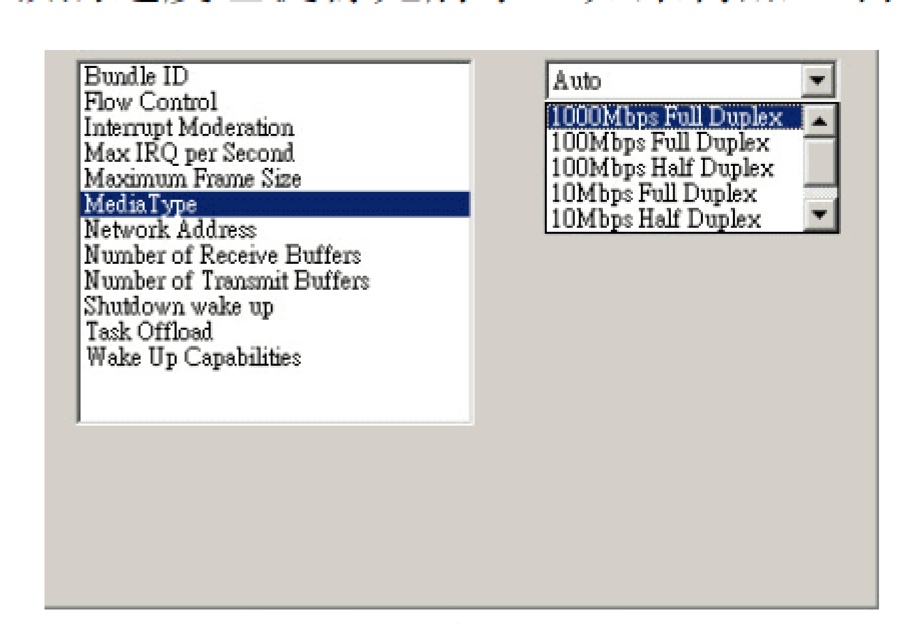
▲ 服务器主板上内置的网卡通常较无兼容性的问题

2. 网卡是否有独立运算芯片

较高级的网卡都有独立运算芯片，以免占用到 CPU 的资源。尤其在 10GB 以上的网络环境，网络提交速度已贴近硬盘访问速度，CPU 开销太多会影响 VM 的使用。一般来说，服务器的网卡都具有此功能，因此只要不是太冷门的产品，读者们可以安心选择服务器即可。

3. 网卡的速度

当前网卡基本都是 GB 速度，但如果 VM 数目多，网卡又不够，一个网卡接口使用多个 VM 是常见的情况，如此一来，多个 VM 如果并发作业，GB 级的速度也捉襟见肘了，如果再加上利用网卡来充当存储界面的 HBA（如 iSCSI，本章稍后会说明），那速度更是会成为“瓶颈”，此时就要考虑 10GB 速度的产品，但此时整个网络环境的交换机及路由器也得升级，会是一笔不小的成本开销。



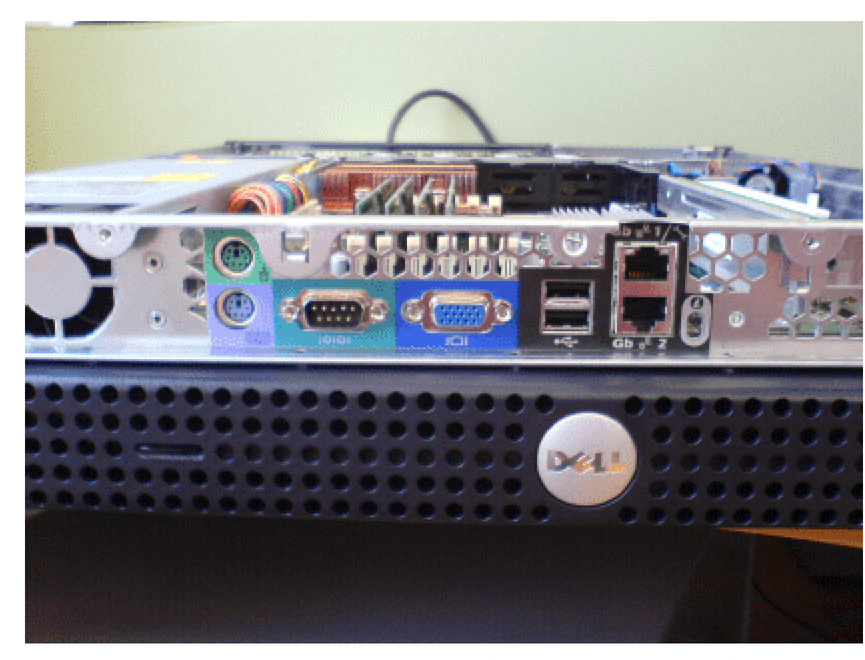
▲ 网卡一般都是使用 Full Duplex 的全双工模式

4. 网卡的任务模式

一般有双工、单工、全双工等。默认值都是全双工，但这些模式本身还是会以网卡的速度为制约，不过具有这些任务模式可以确保在异构网络环境中网卡的兼容性。

5. 内置网络通信端口的数量

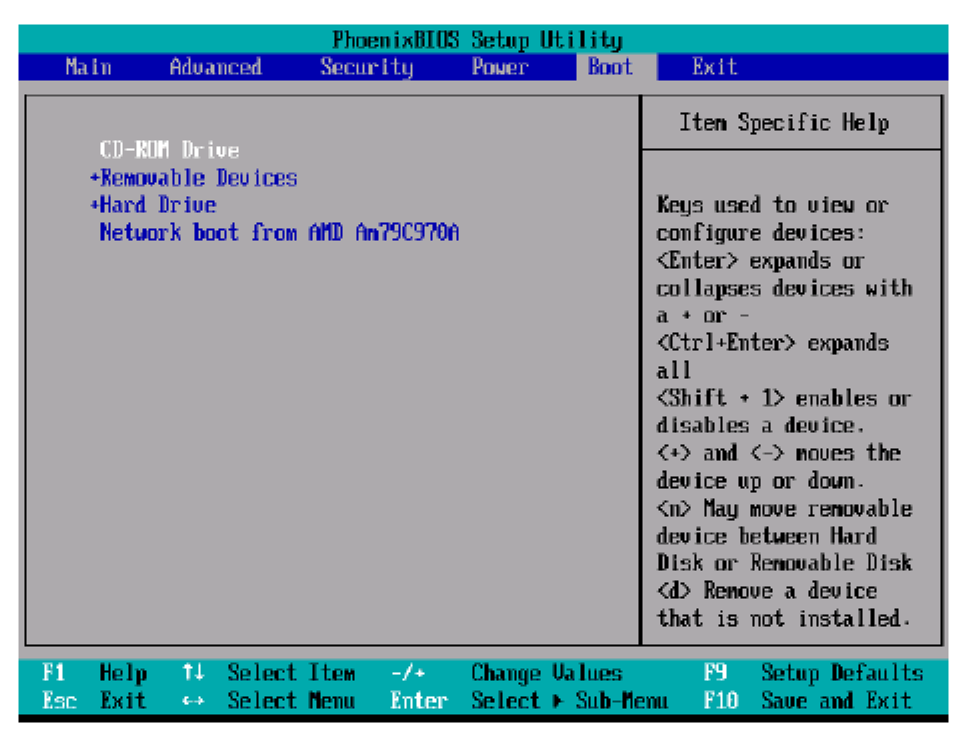
通常服务器内置网络通信端口为两个，较高级的服务器会有四个。当前内置网卡界面大部分为 PCI-E，因此整个 PCI 信道的速度也不会被占满。如果 PCI-E 的速度充足，笔者建议是越多通信端口越好。



▲ 大部分的服务器内置两个网络通信端口

6. 支持 PXE 引导及 BOOT ROM 补丁

网卡一般会具有 PXE 引导功能，可以让计算机在引导时，通过 DHCP 先取得 IP，再利用 IP 网来进入 SAN 引导，从而成为无磁盘的服务器。因此网卡一定要具有 PXE 引导功能才行。有些网卡的 PXE 引导功能需要通过补丁重写固件，因此必须有 BOOT ROM 的补丁功能，在购买机器时一定要先行询问。



▲ 需要支持 PXE BOOT 才能进入 SAN 引导

7. 是否具有 TOE 功能

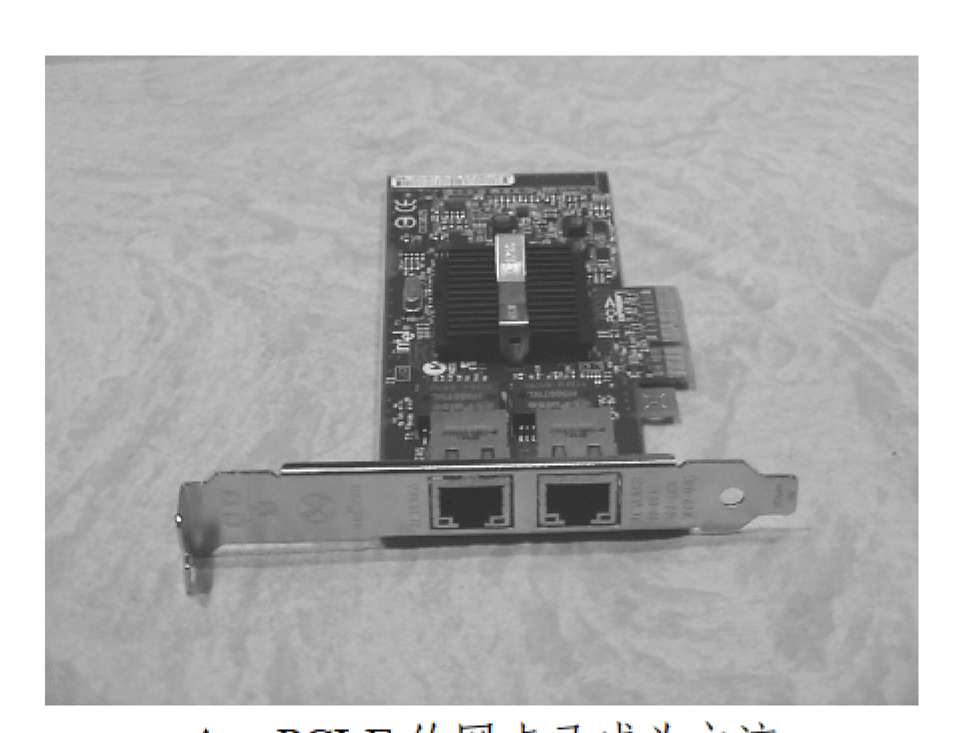
TOE 的全名是 TCP Offload Engine，主要的功能是在将网卡作为 iSCSI 的界面卡时，可以帮助网卡进行运算，可以省下 CPU 的计算时间。TOE 的支持一般是以 TOE KEY 的方式，如果是内置网卡，可以直接插在主板上，因此在购买服务器时，一定别忘记主板要支持 TOE KEY。



▲ TOE KEY 是让 iSCSI 不占用 CPU 的重要部件

8. 外置网卡的界面

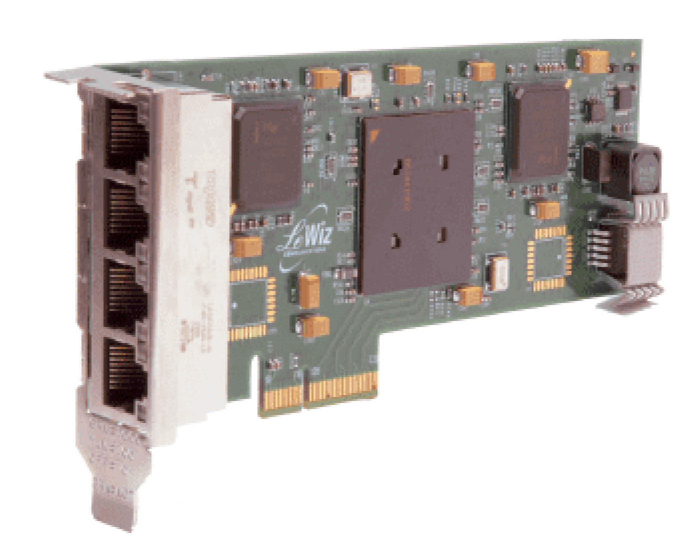
当前服务器的界面均为 PCI-E 或 PCI-X，但笔者建议的是 PCI-E，因为其信道数可以累加，如常见的 1X、4X、16X，因此让界面卡不再受 PCI 信道数目影响速度。如果要购买外置的网卡，一定要给定 PCI-E 界面。



▲ PCI-E 的网卡已成为主流

9. 外置网卡的通信端口数

当前主流的界面都是 PCI-E，而 PCI-E 可以将信道累加，因此单个插槽的速度制约不像从前只有 33MHz，如此一来，单个界面卡上的网络通信端口数也大幅提高，因此我们也在单个 PCI-E 卡上看到四个网络通信端口的产品。笔者十分建议使用多通信端口的网卡，可以分担 VM 的网络通信带宽，但在使用时还是得注意多片网卡的防止单点失误原则，在本书稍后会有介绍。



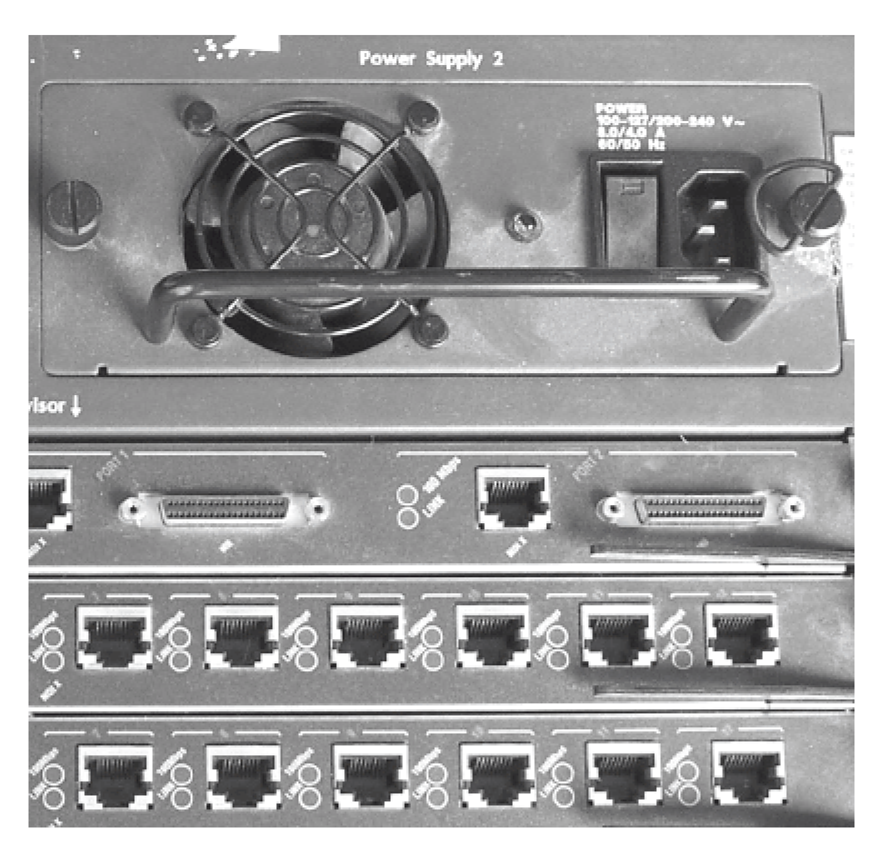
▲ 一卡四通信端口是常见的产品

3.4.2 理解 vSphere 专用的外置网络设备

在一个网络环境中，最重要的就是交换机了。交换机的定义很广泛，包括了路由器、防火墙等都可算是交换机，用来提供机器之间的数据交换，甚至是跨网域的数据交换。vSphere 更支持了虚拟的交换机，这是由 VMware 和 Cisco 联合开发的 Cisco Nexus 1000V，因此在选购交换机时，要特别注意和 VM 配合使用的功能才行，我们就来看看选择外置网络设备的要点。

1. 交换机的速度

交换机才是决定真正网络环境整体效率的重心，就算网卡是 10GB，如果交换机还是 100MB 的，整个网络环境就是 100MB。另外交换机必须是提供每一个通信端口都能确定速度，换句话说，一台 10GB 的交换机必须确保每一个通信端口都能单独运作在 10GB 以上的速度，才是真正提供 10GB 环境的交换机。



▲ 真正的速度必须确定每一通信端口都能有 GB 以上的流量才行

2. 交换机的功能

交换机除了提供标准的分组交换、防火墙、VLAN 等功能之外，还必须包括 vSphere 专用的功能，如通信端口映像、Teaming、Trunking 等。最重要的还是得和 Nexus 1000V 配合。当前交换机厂牌也相当固化，大部分的企业用交换机都具有这些功能，只要在选购之前上 HCL 确定一下即可。



▲ 主要是能支持 vSphere 的 VM 功能，大部分的交换机当前都没问题

3.5 服务器内置存储设备

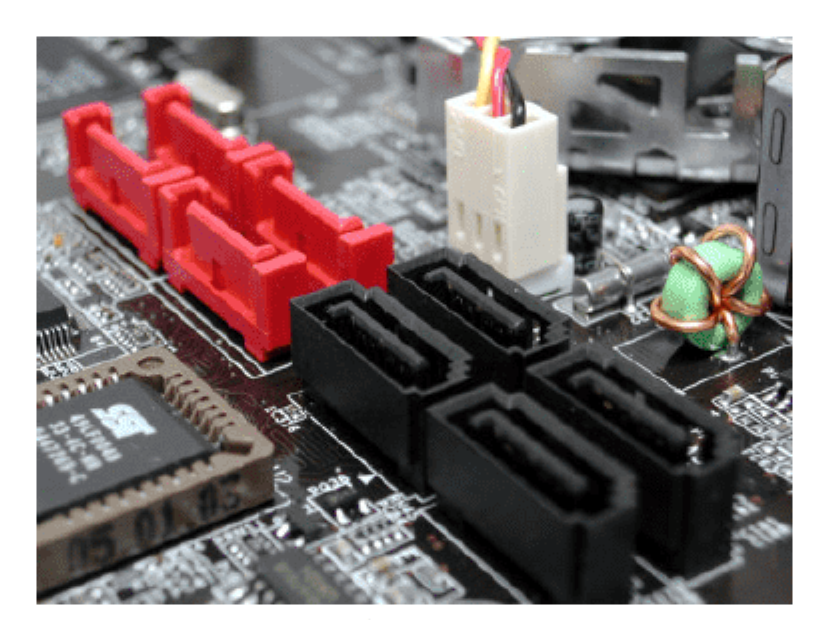
在 vSphere 环境中，物理服务器需要安装一个 Hypervisor 在本身的机器上，通常会是本地硬盘。本地的硬盘选择很多，通常会有不同的界面或是硬盘设备种类，下面就是基本介绍。

3.5.1 内置存储设备的界面

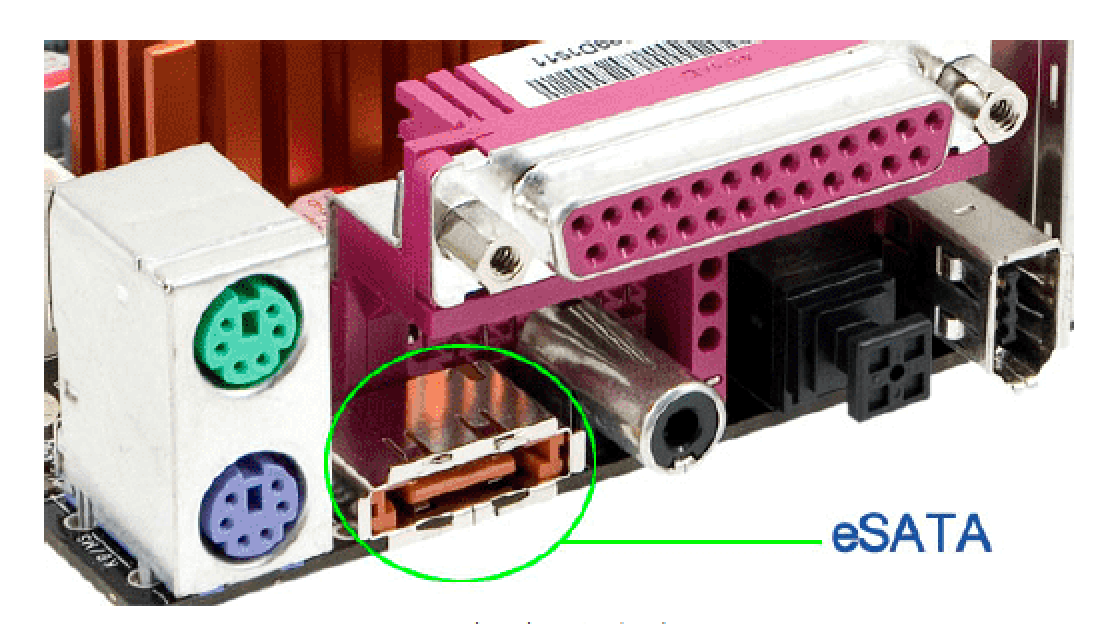
通常服务器的硬盘有 SATA、SCSI（SAS）以及 Fibre Channel 几种，这些界面必须靠主板上的芯片组或是专门的存储芯片来负责，专门的存储芯片包括了主板上附加的，或是外插的存储界面卡两种，这也是 vSphere 会弹出兼容性问题的地方。

1. SATA 界面

SATA 是最常见的硬盘界面，个人计算机中已经普遍使用。SATA 界面存储设备的优点就是取得容易，当前 2TB 的 SATA 硬盘都只要几千元，而 SATA 界面的光盘机也成为主流。SATA 界面的优点是成本低，取得容易，但缺点就是速度和稳定度不如专业的企业产品，但这个缺点可以通过 RAID 来克服。



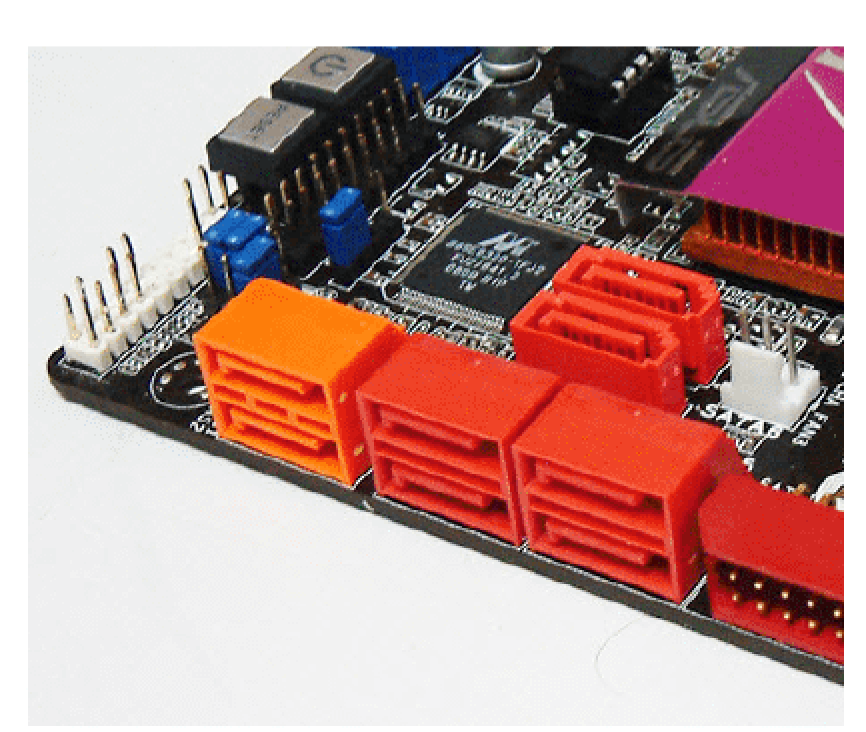
▲ SATA 界面当前已经有 3.0G 的访问速度



▲ 还有外置的产品

2. SCSI 界面

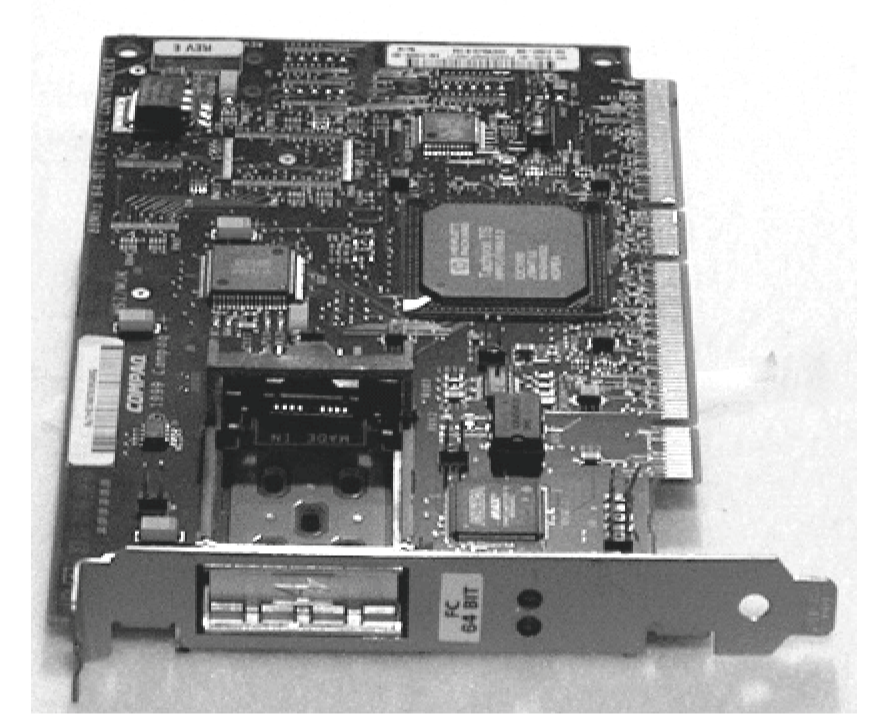
SCSI 界面发展已久，已有 25 年以上的历史，虽然成本高，但其稳定性及高效能一直是服务器的最爱。当前最新的 SCSI 界面为串行式 SCSI（Serial Attached SCSI），其界面和 SATA 完全一样，但能做的排列组合比 SATA 多很多，又可以向下兼容，即 SATA 硬盘能插上 SAS 的界面，因此一直留在服务器上。SAS 界面的稳定性、速度以及并发连接的设备数量比 SATA 强上太多了，但高单价以及取得不容易是 SAS 的缺点。但如果只安装单纯的 Hypervisor，那么 SAS 在成本上的劣势也可忽略不计了。



▲ SCSI 界面当前也和 SATA 一样了

3. Fibre Channel 界面

光缆信道（简称 FC）是一种标准而不是一种界质，使用的材质不一定是光缆，也很少用于内部传输，但少数的服务器还是将 FC 用于内部。FC 的速度奇快，但价钱也特别贵，通常 FC 都用在外部存储上，但当我们将 FC 的光缆线缩短到机壳内，并且利用服务器本身提供电力给 FC 硬盘，那么 FC 也像是内部硬盘了。



▲ 通常 FC 界面是通向外部存储的

3.5.2 硬盘机本身的区别

在存储界面的末端就是硬盘机本身了，也有因转速、机身大小以及访问方式（单纯硬盘或 SSD）而有所区别。由于硬盘机是计算机设备中唯一利用机械运动生成数据的，因此速度和 CPU、内存比起来算是奇慢无比，因此选择不同的硬盘会对 vSphere 的运行有不同的影响。



▲ 硬盘本身也有区别，如 3.5 吋和 2.5 吋的硬盘

1. 硬盘的缓存

硬盘的速度比 CPU/内存慢很多，因此通常是 CPU 和内存在等待硬盘的数据，为了弥补这个时间差，所有的硬盘上都有一个缓存来加快速度。缓存的容量越大越好，当前稍好的 SATA 硬盘都有 32MB 的缓存，而少数企业用的硬盘甚至有 64MB 以上的缓存。

2. 硬盘的转速

标准的 SATA 是 7200rpm，SAS 通常是 10 000rpm 以上，高级的硬盘通常会有 15 000rpm，转速越高速度越快，但通常也会生成高热，价钱也会比同容量低转速的贵上数倍。

3. 硬盘的介质：传统硬盘或 SSD

除了传统的圆盘式硬盘，最近也开始流行固态硬盘（Solid State Disk）。固态硬盘就是不会挥发的内存，在内部使用 RAID 0 的方式存放。SSD 的速度可以比传统的硬盘快上数十倍，但成本也在数十倍左右，但以效能而言，如果仅安装容量不大的 Hypervisor，或是当做 VM 群的交换空间，会让整个 vSphere 的效能大幅提高。



▲ SSD 将会是今后硬盘的主流

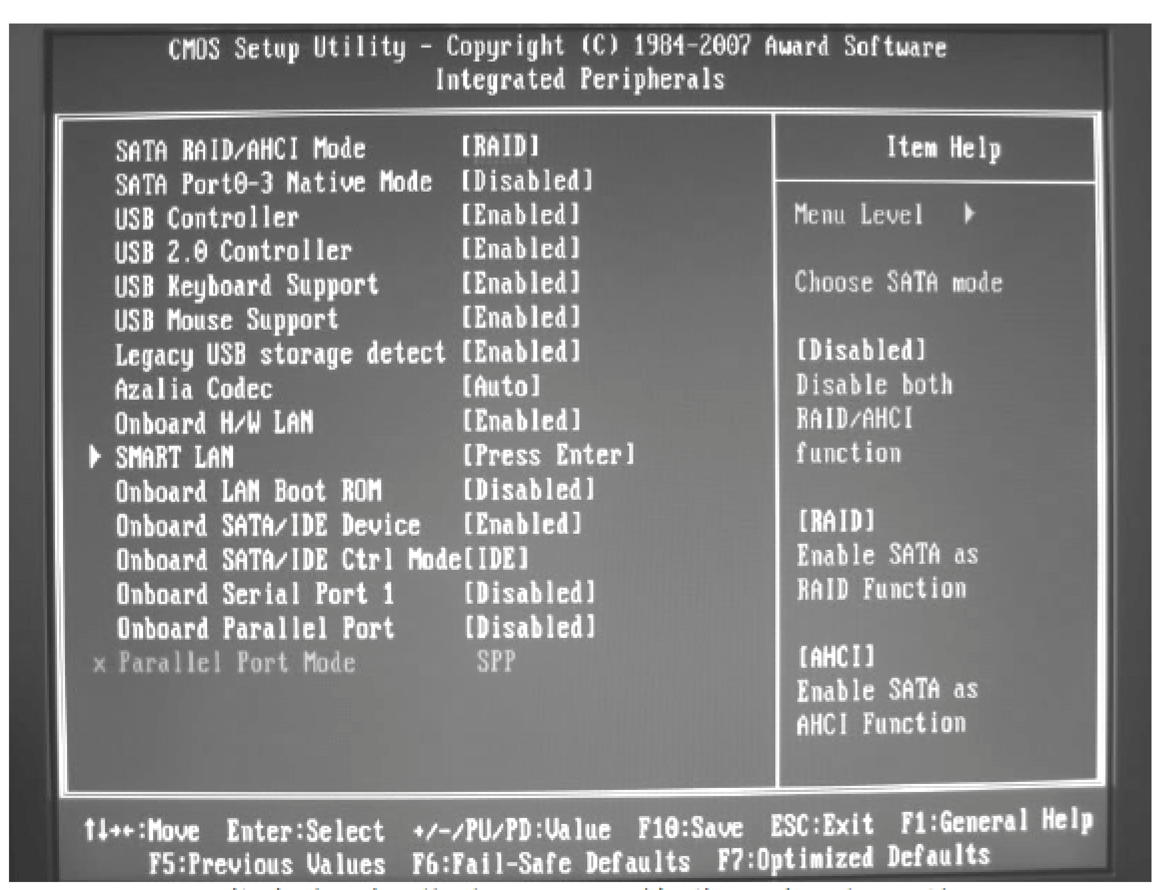
3.5.3 理解什么是 RAID

简单来说，RAID 就是将多个硬盘组合起来，让用户认为只有一个单个超大硬盘的功能。RAID 分成硬件及软件实践，软件一般由操作系统来落实，但在 vSphere 中是不支持软件 RAID 的。

在服务器环境中，硬件 RAID 功能可以由主板芯片组、主板内置的 RAID 芯片或是外置的界面卡来落实，而 RAID 芯片或是界面卡，通常也是 vSphere 弹出兼容性问题的来源。



▲ RAID 主要还是用来将硬盘排列组合

> **注意**
>
> **主板内置 RAID 功能并非真硬件 RAID**
>
> 有些主板上本身就有附 RAID 功能，通常是基于 SATA 界面的，常见的牌子有 Promise、FastTrek 或是 Intel 的 RAID 产品。但这些 RAID 并非真正的硬件 RAID，在使用上并无法达到 RAID 的速度，仅有将不同磁盘排列组合的功能，在数据保护上，仍然需要使用到主机上的 CPU，在此并不建议使用这一类的 RAID 来安装 vSphere。
>
> 
>
>
> ▲ 通常主板上附的 RAID 均非正规的硬件 RAID

1. RAID 的种类

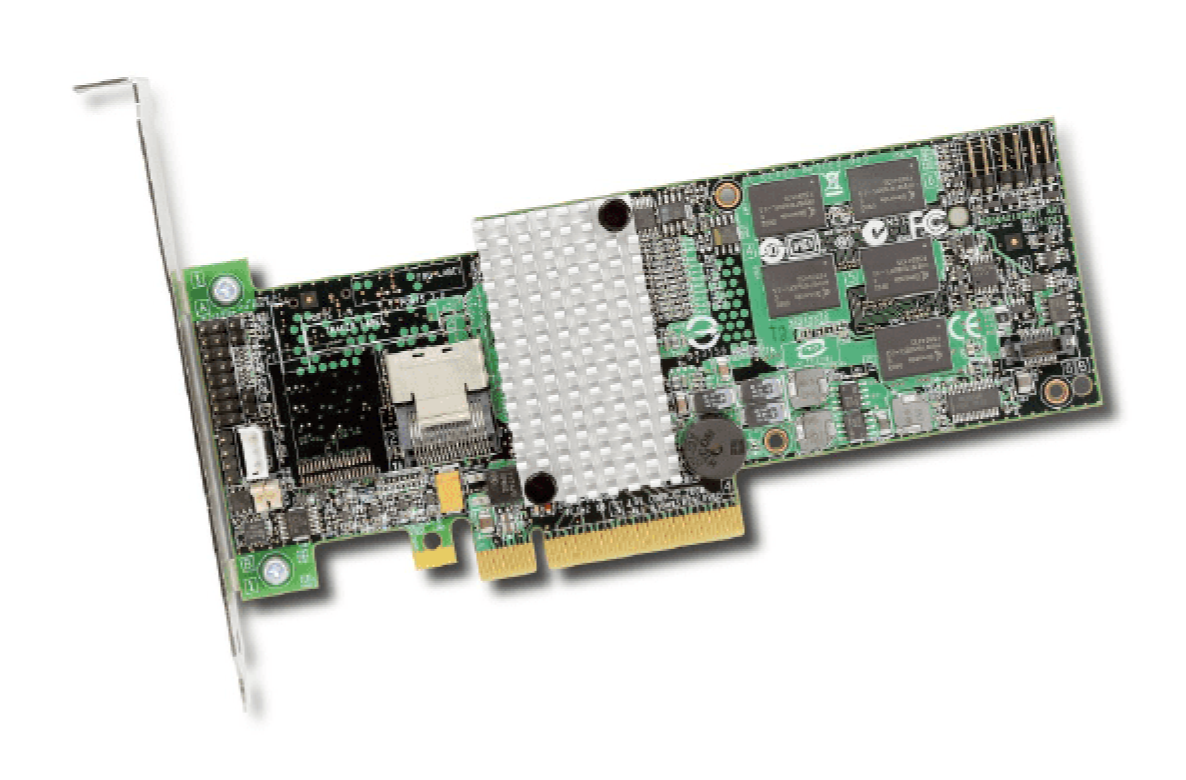
根据不同硬盘的排列组合，RAID 的种类也很多。在 vSphere 环境中，最常见的就是 RAID 1，即将两颗硬盘互相镜像，其中一颗硬盘坏了之后另一颗还能用。另一个常见的就是 RAID 5，至少需要三颗硬盘，总容量是 N-1，允许所有的硬盘坏掉一颗，比 RAID 1 较节省空间，但需要经过数据握手计算，因此速度较慢（见表 2）。

表 2　常见 RAID 种类

	操作手段	功 能	示 例	常用场合
RAID 0	容量：累加 读速度：累加 写速度：累加	无保护仅提速，任何一颗坏掉，所有数据消失	两颗 1TB 硬盘做成一个一颗 2TB 硬盘	不重要的资料，如 Swap 空间或临时文件空间
RAID 1	容量：N/2 读速度：2 倍 写速度：不变	可保护数据，一颗硬盘坏了，数据不受影响	两颗 1TB 硬盘做成一颗 1TB 硬盘	ESX 或 ESXi 的引导硬盘
RAID 5	容量：N-1 读速度：N-1/N 写速度：N-1/N	可保护数据，一颗硬盘坏了不受影响	三颗 1TB 硬盘做成一颗 2TB 的硬盘	不太强调速度，但强调保护的场合
RAID 10	容量：N/2 读速度：累加 写速度：不变	可保护数据又可提速	四颗 TB 硬盘做成 1 颗 2TB 硬盘	速度及数据保护并重的场合

2. RAID 界面卡

一般主板上的 SATA/SAS 界面有限，因此我们在架设 vSphere 环境时，会插上外置的存储界面卡。存储界面卡除了能让主机接上更多的硬盘/光盘之外，最重要的就是提供 RAID 功能，通常在购买服务器产品时，一定会包括 RAID 界面卡。



▲　真正的硬件 RAID 卡功能强大，能做的组合也较多，效能当然更好

3.6　其他要注意的硬件

基本上的硬件介绍完之后，要完成操作一个 vSphere 环境，还需要其他硬件，可帮助 IT 人员更方便地管理企业环境。

3.6.1　远程遥控界面卡

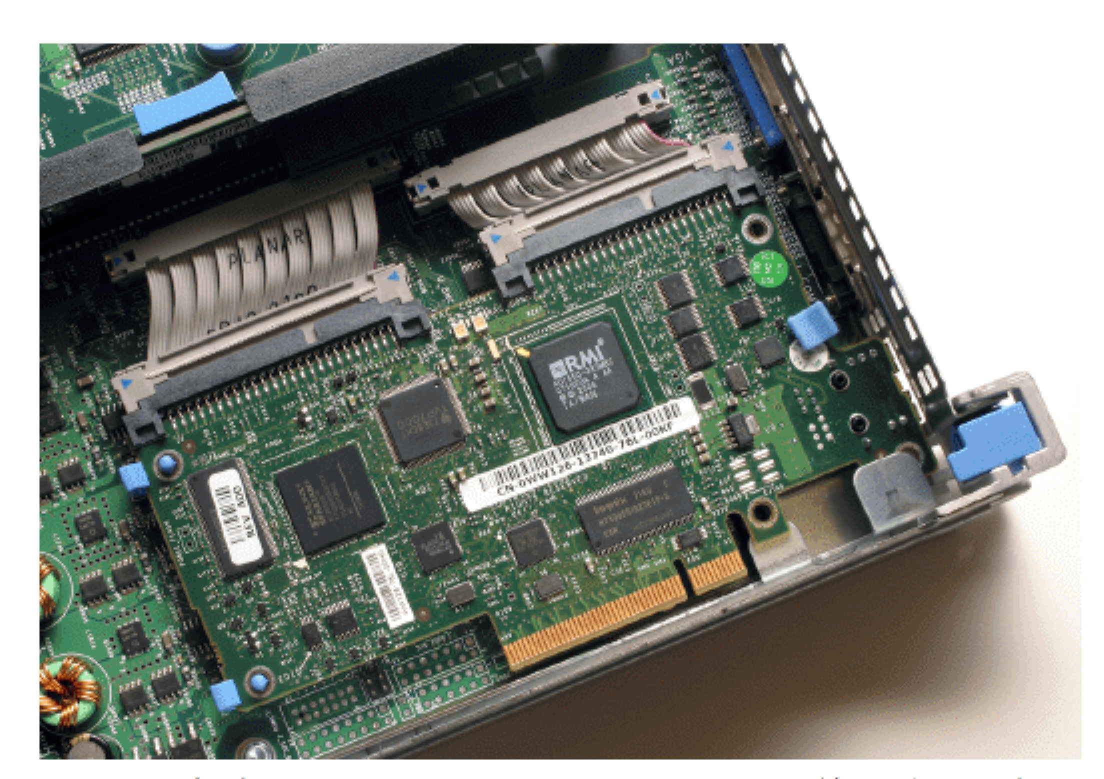
机房环境是严禁任意出入的，但机房内的设备也是常常需要操作，因此通常会使用远程遥控软件来操作物理机或 VM。然而远程遥控软件仅能在引导远程服务装载后使用，并无法通过远程

遥控来配置服务器的 BIOS 或是安装 OS 等。

1. Intel 提出的 IPMI

为解决这个问题，Intel 早在 2001 年就提出了“智慧平台管理界面”（Intelligent Platform Management Interface，IPMI）。简单来说 IPMI 可以通过任何界面，包括 USB、网卡、COM PORT 等来监控系统、服务器在引导时，将所有的屏幕键、鼠标（简称 KVM）导入这个系统，让系统管理员远程接管 KVM，从而总控系统。

当前各大厂家都提出了匹配 IPMI 的产品，如 Dell 的 iDrac、IBM 的 IMM 等，可以直接远程管理 KVM。当前大部分的 IPMI 功能都是通过 IP，也就是说网卡来管理，并且有一个集成式的界面，本书稍后的章节会有完整的介绍。



▲ 此为 Dell iDrac6 Enterprise 远程管理界面卡

2. USB 的 KVM 切换

在非机房环境中，也不可能每一台服务器都有一个 KVM，因此我们会使用多对一的 KVM 切换器。当前均以 USB 界面为主，可以最多到 1 对 8，也就是一套 KVM 对操作 8 台服务器，这一类产品虽然成本不高，但对多台服务器的管理来说可起到事半功倍的效果。



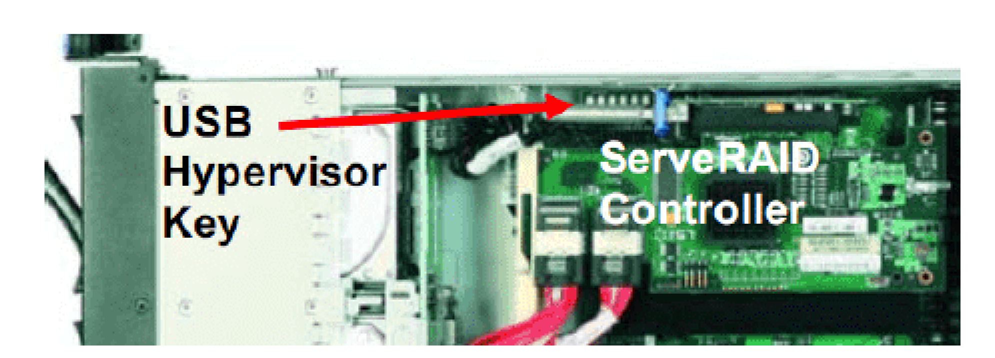
▲ USB 的 KVM 切换在中小型企业也相当实用

3.6.2 内置 Hypervisor 使用的 USB 插槽

在 vSphere 中的 Hypervisor 有 ESX 和 ESXi 两种，而 ESXi 功能规约化，并且允许安装在一个 USB 随身盘中，不但可以加快 Hypervisor 引导的速度，也可以防止硬盘故障时无法使用。有鉴于此，大部分的服务器为了匹配虚拟化的潮流，都特别为 ESXi 准备了一个专门的插槽。

1. 使用 USB KEY 的 ESXi

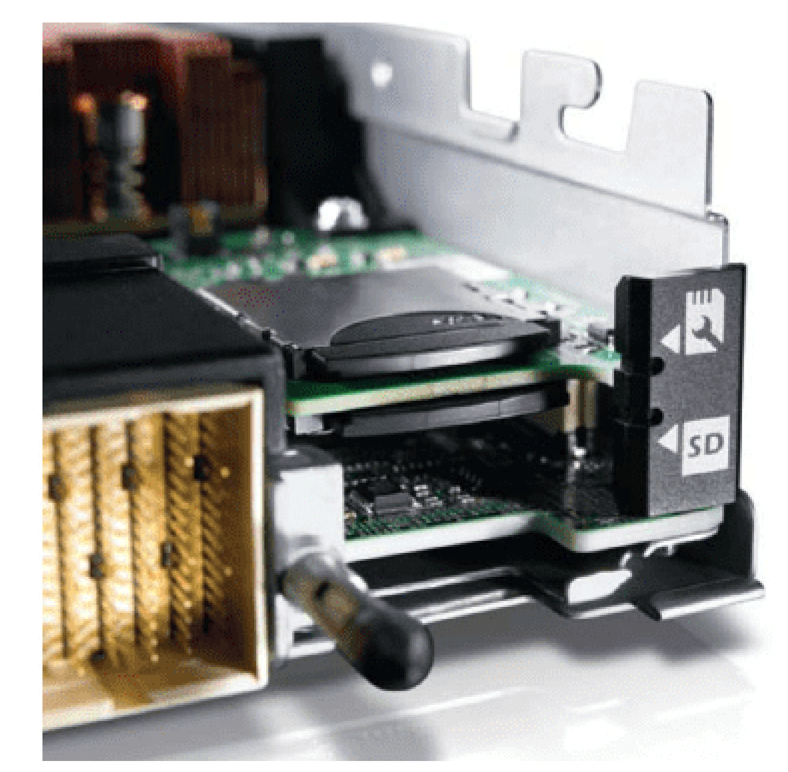
IBM 的主机在其中特别准备了一个 USB KEY 让 ESXi 的 Hypervisor 使用，只要在其中安装好了 USB 版本的 ESXi（本书稍后会界绍），就可以让 USB 中的 ESXi 引导，成为最短小精干的 Hypervisor。你可以随时将这个 USB 放到另一台服务器上引导，成为真正可携式的 vSphere Hypervisor。



▲ IBM 的 USB KEY 是专门用来安装 Hypervisor 的

2. 使用 SD 插槽的 ESXi

Dell 使用了双重 SD 卡来放置 ESXi，可以让服务器从 SD 卡上的 ESXi 引导。Dell 的产品为了防止 SD 蓝屏，特别使用了三个 SD 卡，换句话说，可以通过这三个 SD 卡作为互相备援的 ESXi Hypervisor，更可保障服务器的可用性。我们在本书稍后也会有完整的介绍。



▲ Dell 的服务器也有专门安装 Hypervisor 的 SD 卡

3.6.3 推荐硬件示例

以下为推荐硬件的示例，读者可作为引用。

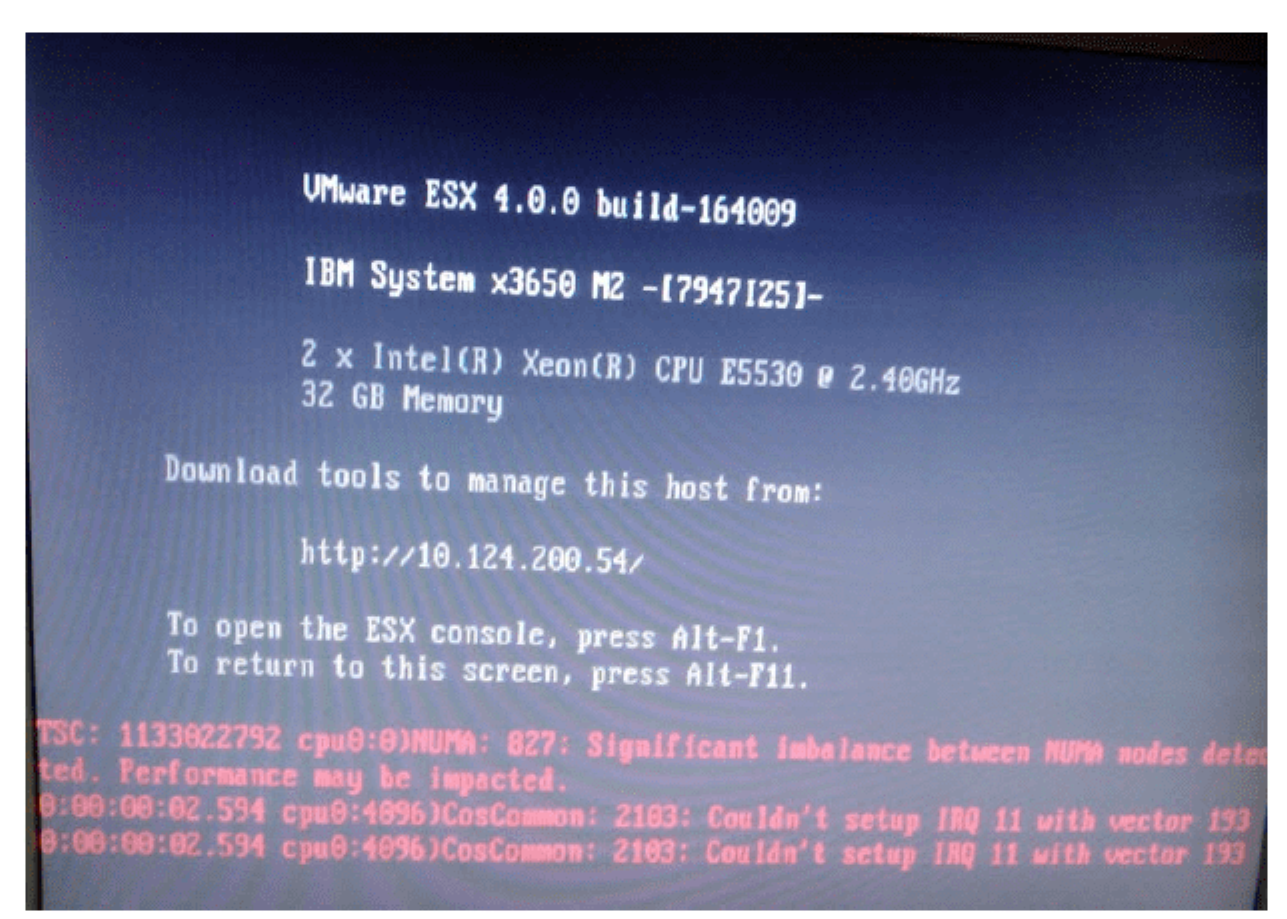
ESX/ESXi 运行推荐硬件

vSphere 服务器	Dell R900，IBM X3550，HP DL380
CPU	Intel 7500/7400 系列，Intel E5500 系列
CPU 数量	至少 2CPU，4 内核共 8 内核
芯片组/主板	Intel E5520 系列
内存	至少 16GB，DDR3，1333MHz 三信道
内存插槽	16 个
内存总数量	至少 128GB
网卡	内置四通信端口，至少 1GB，Intel 或 Broadcomm 芯片带 TOE 功能
外置网卡	Intel Pro/1000MT Server Adapter Quad Port
交换机	Cisco Nexus 系列或 Juniper EX 系列
PCI-E 插槽数	4 个以上
存储界面卡	Dell Perc6i，IBM ServerRAID
硬盘	6 个 146GB SAS 15K RPM 外加一个 60GB 的 SSD
远程遥控界面	Dell iDrac 6 或 IBM IMM 2.0
内置 Hypervisor	Dell SD 或 IBM USB KEY

3.7 个人计算机安装 vSphere

在本节开始前，笔者再三强调，如果要将 vSphere 用在正规的场合，千万不要使用个人计算机来安装 ESX/ESXi。很可能你花上几天或是几个月的时间都无法落实功能，把这个时间成本去买台正规的服务器都较划得来。

如果你真的想安装一个 ESX 来测试，当前 VMware 的 Workstation 版本允许你在其中安装 ESX 的虚拟机，并且在其中激活巢状的 VM，虽然效能很差，但至少基本的功能都可以运行。但如果你真的具有实验精神，并且有不能运行的心理准备，就可以试试在个人计算机上安装 ESX/ESXi。



▲ 在 PC 上安装 ESX/ESXi 遇到不兼容的概率相当高

3.7.1 先到 Whitebox List 检视

全世界的 vSphere 玩家也在不断实验在 PC 上安装 ESX/ESXi，并且将实验成功的硬设备制成清单以供日后其他玩家能引用安装。这个称为“白盒清单”（Whitelist box）的 HCL，是 PC 玩家最重要的引用网站，如果你想用个人计算机实验，就要去 http://www.vm-help.com//esx40i/esx40_whitebox_HCL.php检视。

[Home] [Forum]

Motherboards and unsupported servers that work with ESX 4.0 and / or ESXi 4.0 Installable
Lasted updated - 2010.04.07

Go to Servers and PCs Motherboards Storage Controllers Network Cards Storage

This table contains a list of systems, motherboards, storage controllers and network cards that have been tested and found to wo
Installable. **Please check out the source column for a system you are considering using as some systems will include sp
ESX / ESXi to run on that system or have other issues to overcome.** None of these systems are supported by VMWare for r
those systems please see this link. In same cases for ESXi, the system may be booting from a USB flash drive and be using no l
whitebox systems that will work with ESX 3.5 and ESXi 3.5 please see this list.

If you have tested a system that is not listed here, please post a comment in the forum.

**NOTE: this list includes a number of SATA controllers that provide RAID functionaly via a software component in the dri
controller. Examples would be the Intel ICH series and the nVidia MCP series. ESX 4.0 and ESXi 4.0 do not support tha
functionality thus you will only be able to access the individual drives connected to controllers such as these.**

Other whitebox lists: VMware's Communities List http://ultimatewhitebox.com/ Wikipedia ESX Whitebox HCL

An OK in green indicates that ESX(i) installed without issues. An OK in orange indicates that a work around was required. Please check th

Servers / PCs

Dell Dimension 5150
Dell Dimension 8400
Dell Dimension 9100
Dell Inspiron 530
Dell Inspiron 545

▲ “白盒子清单”可以找到玩家成功安装的硬件配备

除了这个白盒子之外，也有以下几个白盒清单：

▶ VMware 的白盒硬件兼容清单

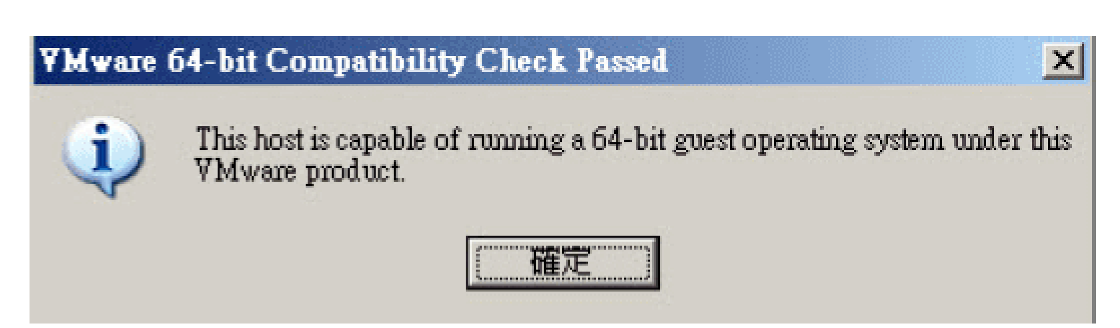
- VMware 小区：http://communities.vmware.com/cshwsw.jspa
- 超级白盒：http://ultimatewhitebox.com/

3.7.2 在个人计算机上安装失败的主因

由于 ESX/ESXi 也是从 RHEL 修改的 Linux，因此基本上能运行 RHEL 的硬件应该都可以安装 ESX/ESXi。但由于 ESX/ESXi 是一个 Hypervisor，因此在外置界面卡上都十分严谨，因此主板的芯片组，网络界面卡及存储界面卡是最常发生兼容性问题的关键。

1. CPU 不相容

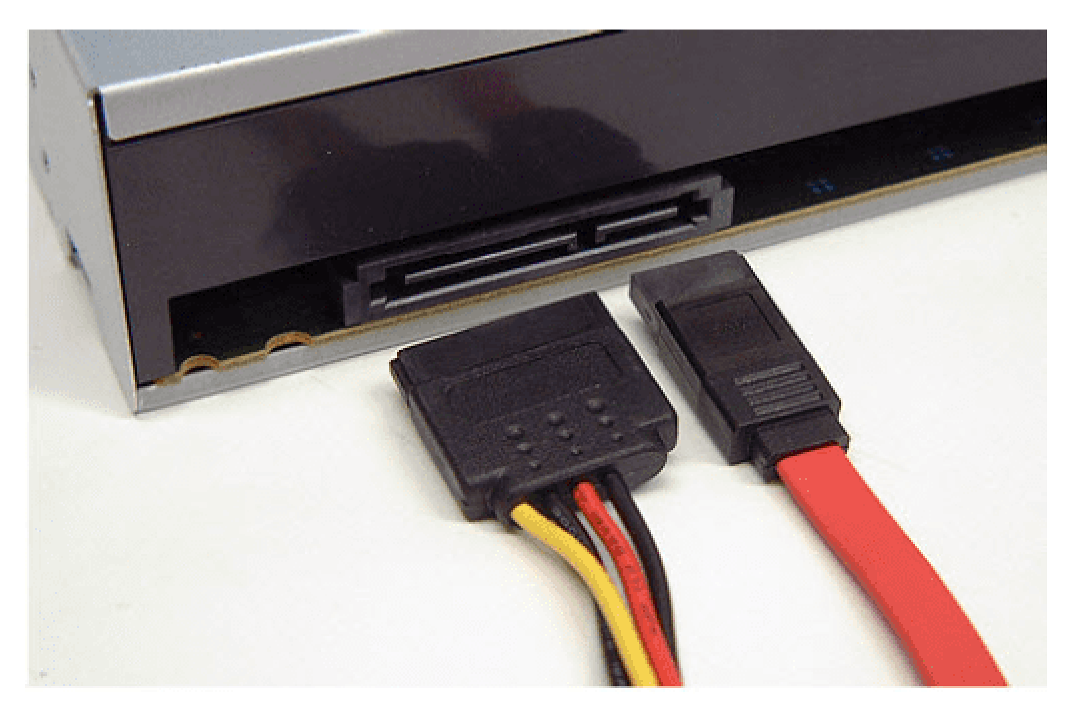
个人计算机使用的 CPU 大部分非 Xeon 系列，因此能安装 ESX/ESXi 的风险就变大了。最常见的就是不支持 VT-X，或是非 64 比特，或是不支持 SSE4 命令。这些都会让 PC 根本无法安装 ESX/ESXi。笔者的经验是，个人计算机中，Celeron 等级的 CPU 通常都会有问题，而双核以上的 CPU 则可以运行的概率较大。



▲ 可以使用 VMware 提供的 64 比特遍历工具来判断

2. 光盘机

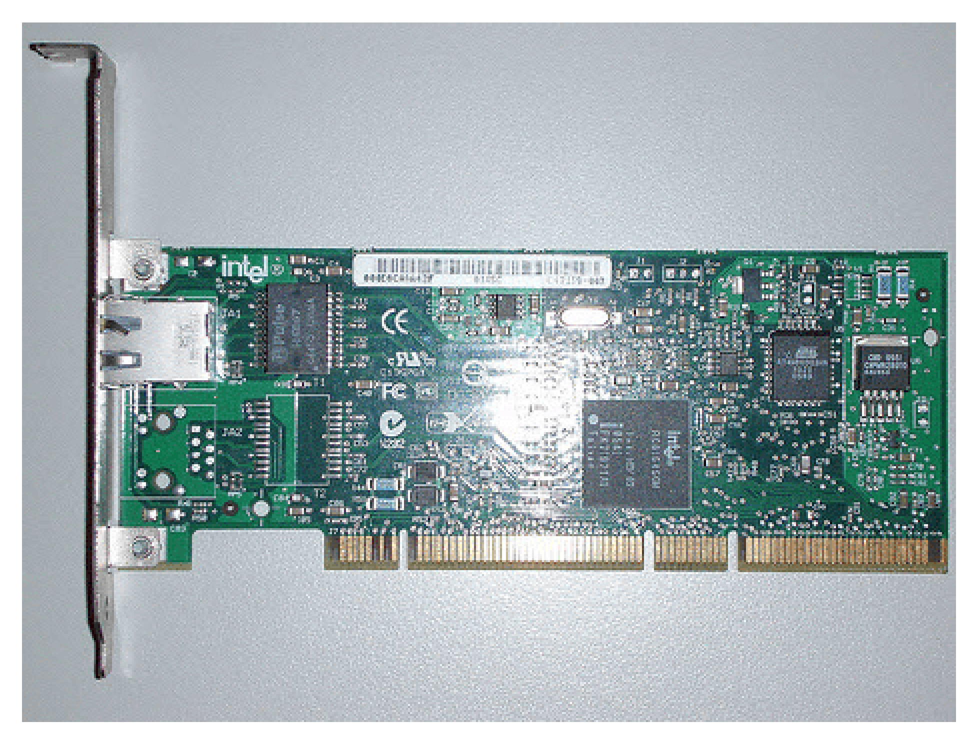
安装 ESX/ESXi 需要用光盘机，但在 2006 年后，PC 的芯片组已经将 IDE 界面卸除，厂家为了兼容性，通常会安装一个额外的芯片组来驱动 IDE 界面，而大部分光盘机也是 IDE 界面。这额外的芯片组通常不会被 ESX/ESXi 辨识，因此在光盘机无法抓到，就无法安装 ESX/ESXi 了。此时使用 SATA 或 USB 界面的光盘机通常可以成功，因此在购买光盘机时，不要再买旧式 IDE 界面的了。



▲ 使用 SATA 的光盘机安装成功概率较大

3. 网卡

网卡是 ESX/ESXi 最挑的硬件，市面上 90%的 PC 网卡都无法使用。因此建议使用 Intel 或是 Broadcomm 芯片的网卡，读者们可以到官方的 HCL 或白盒 HCL 来检视 ESX/ESXi 支持的个人计算机用网卡。



▲ 网卡就买 Intel 或 Broadcomm 的

4. 内置 SATA RAID 很少成功

RAID 卡也是 ESX/ESXi 最挑的一部分，可以说 PC 上 99%的 RAID 卡都无法使用。由于 Intel 在个人计算机上早就有内置 SATA RAID，因此很多人都想将 ESX/ESXi 安装到 RAID 上。但大部分的 Intel RAID 都无法成功，因为这些 RAID 都非真正的硬件 RAID，只能做到将 SATA 硬盘排列组合而已，因此在 PC 上安装 ESX/ESXi 时，千万不要创建 RAID。

```
Intel(R) Matrix Storage Manager option ROM v7.5.0.1017 ICH9R wRAID5
       Intel(R) Matrix Storage Manager option ROM v7.5.0.1017 ICH9R wRAID5
          Copyright(C) 2003-0-----[ MAIN MENU ]----All Rights Reserved.
/---------1.  Create RAID Volume--[ MAIN MENU3.  Reset Disks to Non-RAID-------\
|         2.  Delete RAID Volume              4.  Exit                         |
|                                                                              |
|                                                                              |
\------------------------------------------------------------------------------/
/--------------------------[ DISK/VOLUME INFORMATION ]-------------------------\
/ RAID Volumes:                                                                \
| ID   Name              Level              Strip  Size     Status      Bootable |
| 0    Test              RAID0(Stripe)      128KB  149.1GB  Normal        Yes    |
|                                                                              |
| Physical Disks:                                                              |
| Port Drive Model       Serial #                  Size     Type/Status(Vol ID) |
| 0    ST380817AS        4MR407JH                  74.5GB   Member Disk(0)     |
| 1    ST380013AS        4MR0DVAF                  74.5GB   Member Disk(0)     |
| 1                                                                            |
|                                                                              |
|                                                                              |
|                                                                              |
|                                                                              |
|              -Select        [ESC]-Exit          [ENTER]-Select Menu          |
```

▲ 在安装时，千万不要使用主板内置的 RAID

结 语

理解硬设备的规格是安装 ESX/ESXi 的基本步骤，在效能和管理之前，成功安装 ESX/ESXi 是必要的条件。当然为了之后的扩展性也相当重要。在使用虚拟化的架构，不需要担心硬件本身的升级能力，因为可以用多台服务器来扩展资源，因此只要掌握了硬件的兼容性，vSphere 的硬件架构也就几乎落实了。

读书笔记

第 4 章 vSphere 的基石——存储设备

关键词：

- DAS
- NAS
- SAN
- Fibre Channel
- SCSI
- iSCSI
- LUN
- TOE
- HBA

存储简单来说就是存放数据的空间，如硬盘、光盘、磁带机等。但随着计算机技术的进步，存储也成了一门独立的学问。在 vSphere 环境中，如果没有存储的支持，充其量只不过是一台能安装多个 OS 的服务器，但在加上了存储的设备后，vSphere 竟然可以担当整个 IT 环境的基础架构，我们在这一章就来看看独立存储设备对 vSphere 环境的重要性。

4.1 外置存储设备的重要性

外置存储设备并不是什么新的技术或应用，在几十年前计算机还不普及时，就已经是政府机构或是教育单位的标准配备了。然而 X86 的计算机一开始就以个人使用为主，因此在计算机中放置一个永久性的存储设备（就是硬盘）本来就是为了方便，但却没有考虑到故障会起到的问题，而这种手段在服务器以及网络开始流行之后，对于多人使用的企业环境竟然变成了一个负担。



▲ 外置存储设备已进入家用领域，图为支持 RAID0/1 的 NAS

4.1.1 内部存储设备的问题

对企业环境来说，数据其实才是计算机操作的热点。数据被存放在非挥发性的介质中（相对于内存的挥发性），在没有电力的情况下还能永久保留，随时随地可供用户访问。数据本身、运算逻辑及界面，三者创建了整个系统，而在这整个信息系统的运作底层，最重要的就是提供一个不会让这三者陷入危险的环境。



▲ 从存储的南桥到扁平电缆及硬盘，任何一个故障都会影响整机使用

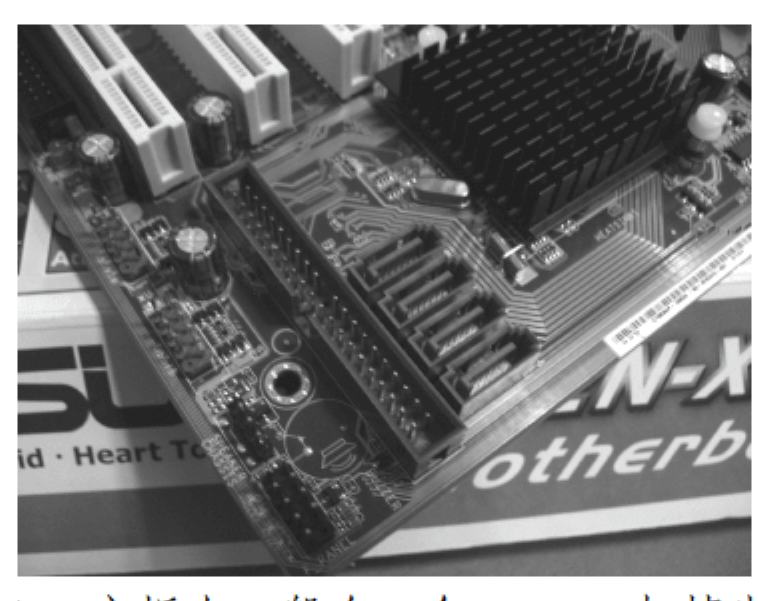
大家对于整组计算机的概念就是快捷的 CPU、大量的内存和一个可以存储数据的硬盘，少了其中一个部件的计算机就是不完整的。但这种组合的计算机，却存在不少问题。

1. 限制性较大

硬盘依附在计算机上最大的问题就是其天生的限制，这些限制包括了：

▶ 内部存储设备的先天限制

- 计算机机壳的大小。
- 电源供应的瓦数。
- 计算机主板上或存储扩展卡能接上硬盘的数量。
- 计算机通过主板和其他计算机提交数据的速度。
- 操作系统能支持硬盘的数量。
- 操作系统能支持文件系统的大小。



▲ 主板上一般有 4 个 SATA，扣掉光盘只剩 3 个，仅能满足个人使用

2. 排列组合的弹性较小

一台计算机中的硬盘访问或配置，是运行在其上的操作系统总控，因此受到了操作系统本身的限制，这些硬盘在最底层的配置或规划，就只能通过操作系统来运作，无法落实一些低级的配置和应用，包括了：

▶ 内置硬盘的弹性

- 无法安装多个文件系统。
- 无法安装多种不同格式硬盘（如 SCSI 和 SATA 结合）。

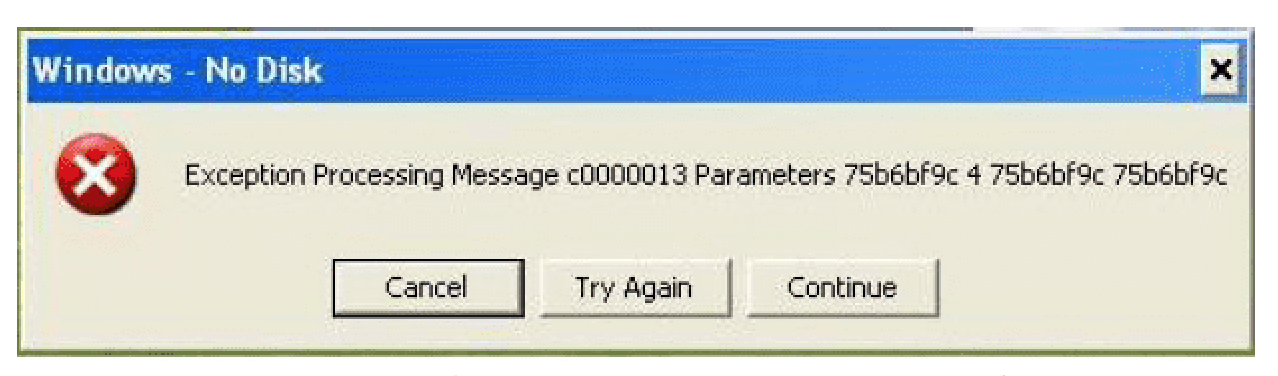
- 无法让所有硬盘共享一个连外信道（如高速光缆）。
- 无法在硬盘空间不够时，在不关机情况下新添加容量。
- 无法做高级的 RAID 组合。
- 无法给定底层的共享。
- 无法做成低级的 Volume Group。



▲ 通常内置 SATA 的 RAID 功能都很简单，并且也非真正的 RAID

3. 单点故障就完全停机

数据放在硬盘中，CPU 和内存及其他硬件（电源、主板、声卡、显示适配器）运作引导之后，操作系统提供用户一个任务环境来访问硬盘中的数据。问题是，只要这整个系统的任何一个地方弹出故障，这台计算机就蓝屏了。计算机蓝屏时，不要说其他通过网络分享这台计算机硬盘数据的用户了，连本地用户都无法再访问数据。如果这台计算机的用途只是让众多用户放置数据的文件服务器，为什么要将这么多不确定的因素全部加诸在这台计算机上呢？为什么不能提供一个单纯的文件，甚至是硬盘的存放环境？



▲ 当硬盘弹出问题时，整个系统就停机了

4. 无法适应集群的环境

基本上当我们在做冗余的集群的环境时，一台计算机发生蓝屏无法使用，另一台待命的计算机必须在不影响服务常规运作的时间内，立即接替原来计算机的任务。举例来说，在做数据库的冗余时，一台 SQL 蓝屏之后，另一台同样运行完全一样进程（Instance）的 SQL 服务器必须立即接手。虽然计算机的服务没有停止，待命机器也顺利将任务接手，但原来服务器中，SQL 数据库如果随着那台机器的蓝屏一起关机，那硬盘中的数据就无法被新的计算机可读，接手之后的新计算机，其上的 SQL 服务就算运行常规，还是会有数据消失的问题。



▲ 只有独立存储设备才能做到 Active-Active 的集群配置

正确的做法，当然是 A 服务器将数据可写一个独立的“硬盘”中，B 服务器则拥有访问这台独立“硬盘”的权限。当 A 蓝屏时，B 立马接手，并且开始访问这台“硬盘”。原来 A 可写的数据 B 也可以访问，在这种情况下，这台“硬盘”就是一个外部的独立存储设备。

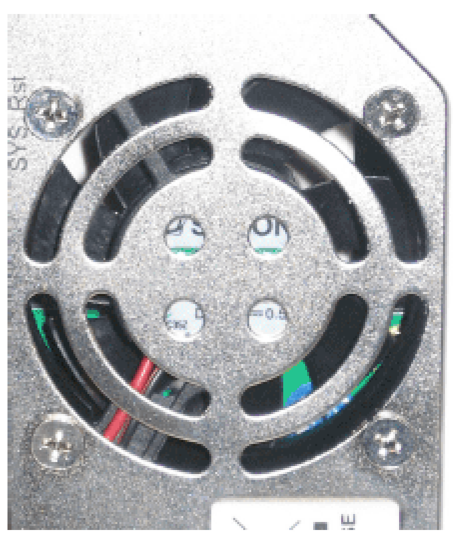
4.1.2 外置存储设备简介

为了解决内部存储设备与生俱来的问题，不是让多台服务器连接到一个非本地界面卡或主板连接设备的“硬盘”，而是利用其他方式 （如专属界面卡或直接走网络），连接到一个外部的“硬盘”。这个硬盘拥有自身的电源系统，不需要依赖一个“完整”的操作系统才能运作，更可以有多种文件格式、能容纳多种异构的硬盘（如 SCSI、SATA、SAS 共存），并且本身已经有防

灾的机制（如 RAID 5）等。这就是外置存储设备的概念。

1. 拥有自身的生态系统（EcoSystem）

外置存储不需依赖任何操作系统或是硬件架构，其拥有自身的电源、处理器、内存、防单点失误（本书稍后会说明何谓单点失误）的外部总线、可供用户配置的界面等。只要是能连入这个外置存储的其他设备，都可可读其上的数据。



▲ 独立存储设备拥有自身的电源及系统

2. 多种硬盘界面

可以让不同界面的硬盘放入同一个硬盘柜，并且允许热插拔，在硬盘弹出故障或是容量不够时，可以随时加入新的硬盘。



▲ 拥有两个插槽的小型 NAS

3. 多种防护机制

一个标准的外置存储甚至可以放入成百上千个硬盘，因此一般的 RAID 可能都不够用了。不过以标准的存储系统来说，使用 RAID 5/6/7 以上的机制可以让整体数据的安全性再度提高。

4. 多种副本机制

大容量的磁盘柜不可能用单个方式来副本。较先进的外置存储设备都有影子系统，可以将任何可写直接副本备份起来，也有实时的差异副本，就是当系统有任何可写动作时，会立即将这个改变的数据副本起来，不同的存储设备厂家都有自身的副本机制。



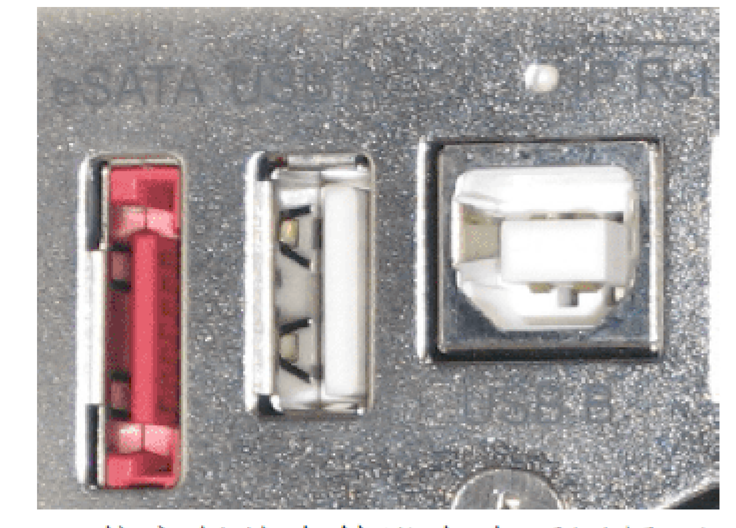
▲ 这是使用 Switch 配置 RAID 的 NAS

5. 多种联机方式

外置存储其他服务器连接的方式不应该受到限制，因为你不会预先知道将连入的服务器或是设备使用的连接方式。因此大部分的存储设备必须支持各种的连入方式，如标准的 SAS、SCSI、Ethernet、高速 Ethernet 或是光缆网络等。

6. 多种文件系统

由于不同的操作系统能支持的文件系统不同，对外置存储来说，必须兼容于不同的文件系统，因此本身即具备支持各种不同文件系统的能力。举例来说，SAN 系统可以将许多硬盘连接起来重新划分，并且给每一个逻辑磁盘不同的文件系统，让 Windows 或 Linux 直接安装在上面，而 VMware 虚拟机最常用的 vmfs 也可以直接安装在外置存储上，下一章我们会详细说明独立存储上的文件系统。



▲ 较高级的存储设备有不同界面，如网络、USB 及 eSATA

```
Disk /dev/sda: 256 MB, 256901120 bytes
255 heads, 63 sectors/track, 31 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1               1          31      248976    b  W95 FAT32

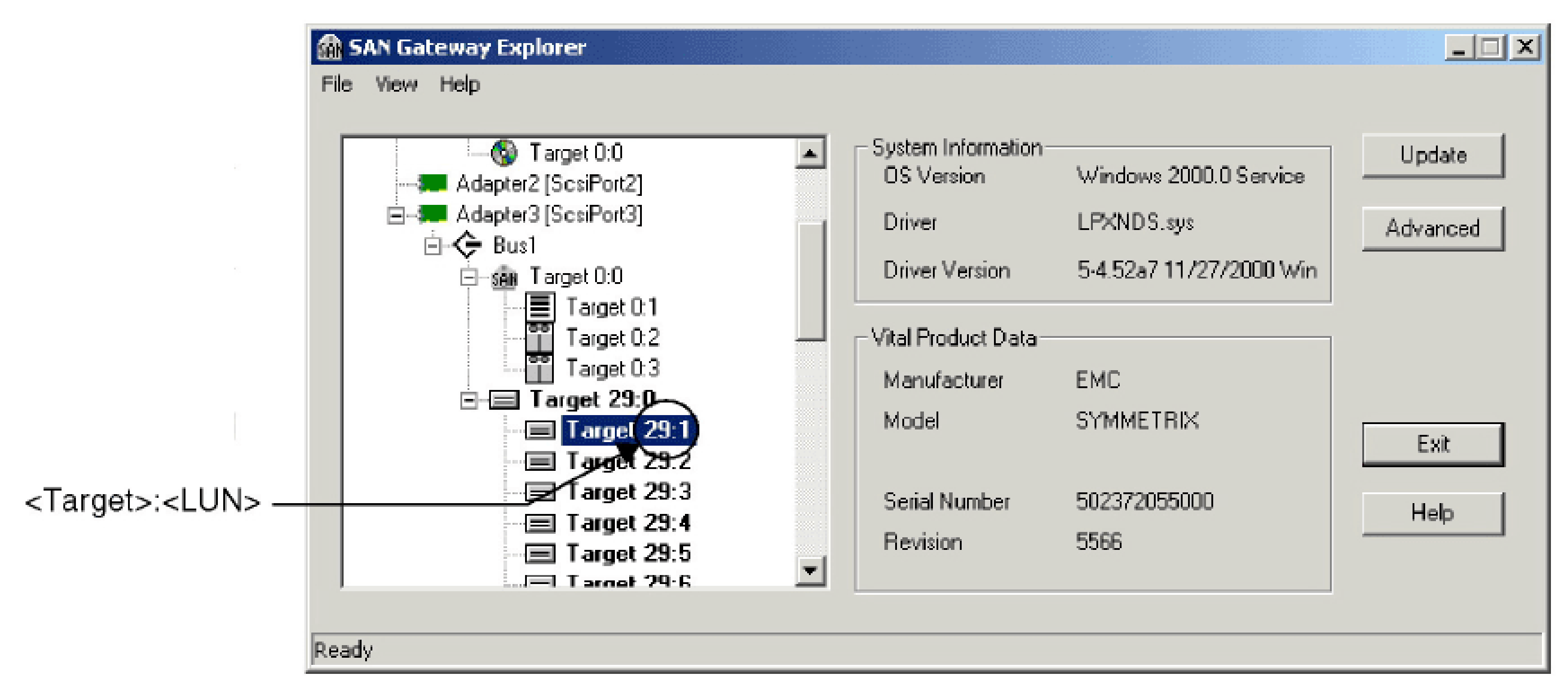
Disk /dev/sdb: 4294 MB, 4294967296 bytes
255 heads, 63 sectors/track, 522 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sdb1   *           1          25      200781   83  Linux
/dev/sdb2              26         522     3992152+   5  Extended
/dev/sdb5              26         217     1542208+  83  Linux
/dev/sdb6             218         299      658633+  83  Linux
/dev/sdb7             300         327      224878+  82  Linux swap / Solaris
/dev/sdb8             328         342      120456   83  Linux
/dev/sdb9             343         522     1445818+  83  Linux
root@tty1[/]#
```

▲ 可以支持多种文件系统

7. 多种访问方式

这里所说的访问方式主要是以传输数据的单位决定。举例来说，NFS 是以卷方式挂载到现存的文件系统之下，NAS 是以文件（File）为主的提交方式，SAN 则可以访问到逻辑单位（LUN），SAS 甚至可以访问到 Block Device（或称单个物理磁盘）。不同的用途使用不同的访问方式，在一个完善的外置存储上，必须提供不同的访问方式。



▲ SAN 可以访问到 LUN，让使用上更有弹性

8. 弹性化的容量管理

5 台 500 GB 的硬盘组合成 2 TB 的 RAID 5 外置存储设备，必须能在其上再划分不同的逻辑磁盘。LUN 的大小从数 MB（给早期的 DOS 用）到数 GB 都有可能，因此外置存储设备必须拥有弹性的空间划分方式以适用于各种不同的场合。

4.2 常用的外置存储设备 DAS、NAS、SAN

外置存储设备是一项单独的科学，绝非短短几页说明可详细描述，因此本章仅介绍和虚拟机最相关的常见产品。事实上以 OSI 七层架构的角度来看，NFS、DAS、NAS 和 SAN 根本不是在同一层次的，在纯技术的对比上毫无意义。但这四者在存储的实用上最常被相提并论，在 vSphere 中也常常被拿来做对比，因此我们在这一小节中，就以 vSphere 使用的角度来看看 DAS、NAS、SAN。

4.2.1　直接连接存储设备（Direct Attached Storage，DAS）

DAS 是最普遍的存储设备，举例来说，你现在计算机上接的硬盘就是 DAS 的一种。顾名思义，直接连接的存储设备，就是和主系统直接相连的。通过一种总线连上主系统之后，主系统上的操作系统可以访问其上的数据，这就是基本的 DAS。当然随着存储技术的进步，DAS 也开始有不同的变化了。

1. DAS 使用的通信协议

最常见的 DAS 通信协议包括了 ATA、SATA、SCSI、SAS 和光缆信道，允许和一台服务器相连，相连的速度则视通信协议而定，一般我们将服务器和 DAS 之间连接的方式称为主机总线界面卡（Host Bus Adapter，HBA）。

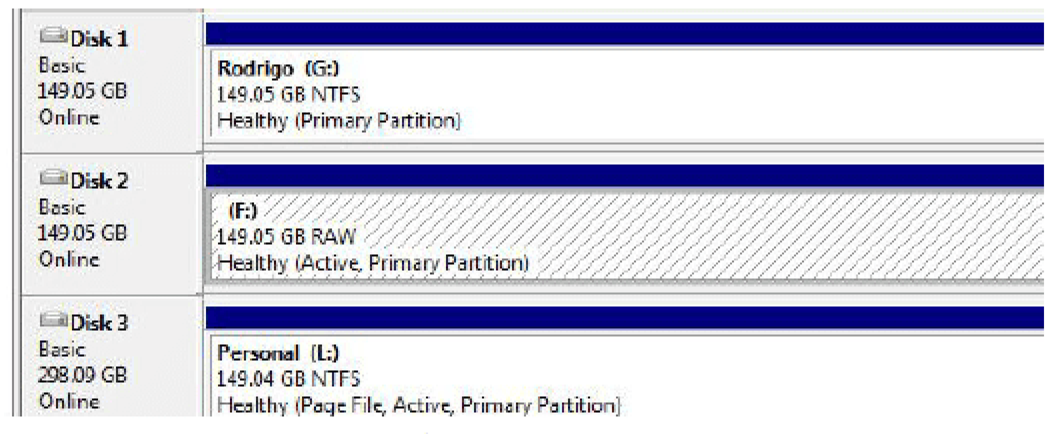
DAS 由于只能被一台机器连接，因此被称为“信息的孤岛”，而 DAS 无法共享的特性更让 NAS 和 SAN 成为服务器的主流，当然 DAS 在大部分的单人或小型企业使用上还是有其优势的。



▲　当连上 USB 时，我们可以视这个设备为 DAS 的一种

2. DAS 的配置

DAS 一般在引导时就会被操作系统辨识出来，因此会在设备管理器中安装驱动程序来管理，会被视为一个本地磁盘，使用标准的操作系统配置设备来操作，如 Windows 中的磁盘管理或是 Linux 下的 fdisk。



▲　DAS 被视为一个本地磁盘

3. 不同等级的 DAS

在 NAS 和 SAN 流行之后，DAS 用途获取了更进一步的延伸。当前在服务器上的 DAS 已不再只是本地硬盘了。而是具有冗余、总控器冗余、冷却冗余以及 RAID 冗余的多功能存储机器，而入门的 DAS 系统则将这些冗余系统内置在 HBA 上，如主板内置 ICH7/8/9R 的 Intel 芯片组就提供了 RAID 0/1/10/5 的功能。

中高等级的 DAS 则提供了内置的总控器，并且拥有自带的 RAID 功能，因此不需要由 HBA 来提供 RAID 功能。较高级的 DAS 也允许共享，让不同的服务器能访问同一个逻辑存储单元，但一般只用在集群上，因此高级 DAS 的功能和入门的 SAN 也没有太大的差别。

> **提示**
> **DAS 连接的主机数量**
> 依规定来说，连上 DAS 的主机只能有一台，如果你通过同样的方式将这个 DAS 设备和第二台主机相连时，这个 DAS 设备就可以被称为 NAS 了。举例来说，有些外置式硬盘使用 USB 来连接，这个 USB 硬盘就可以被视为一个 DAS 设备。但如果这个外置式硬盘有两个 USB 端口，并且让不同的两个主机并发利用两个 USB 端口连入，并且可以并发读写数据，那么这台外置式的 USB 硬盘，在较不严谨的认定下，必须被视为一台 NAS 或是 SAN，连接的总线是 USB 端口。

4. 一些 DAS 设备示例

DAS 的设备主要还是接在一台计算机上，下面就是一些标准的 DAS 示例。

- USB 硬盘
- 单机光盘机
- 各种接在计算机上的硬盘（IDE、SATA、SCSI、SAS 界面）
- 磁带机
- JBOD 磁盘数组
- 外置 RAID 磁盘机
- 非网络服务器延伸存储设备

4.2.2 网络连接存储设备（Network Attached Storage，NAS）

NAS 是大家最常听到的网络存储设备。从其名称上来看，就是接到“网络”上的存储设备。以当前主流的网络来说，就是 Ethernet 网络，当然在加上了互联网之后，就是 TCP/IP 网络。

1. NAS 的定义

NAS 的定义较简单，其为以“文件”为主的共享存储设备，通过“计算机”网络和服务器之间进行数据交换，因此我们在 NAS 设备上看不到 LUN 或是整个文件系统。另外只要是网络协议都可以用在 NAS 上，因此 NAS 可以在各种不同的网卡共享数据。



▲ NAS 是使用网络连接的

2. NAS 本身的机器

NAS 本身其实就是一台计算机，在硬件上，有主板、CPU、内存、I/O 设备等，软件上则有操作系统和应用软件，但操作系统和应用软件的目的都是提供文件共享及文档系统和其相关的安全、管理等功能。用户当然可以找一台一般的服务器，并且安装上 Windows Server 或是 Linux 操作系统，并且开放文件共享，这台计算机当然也就成为一台 NAS，但如果这台计算机在存储设备上的功能不够强，就无法成为一台称职的 NAS 了。

3. NAS 的磁盘本体

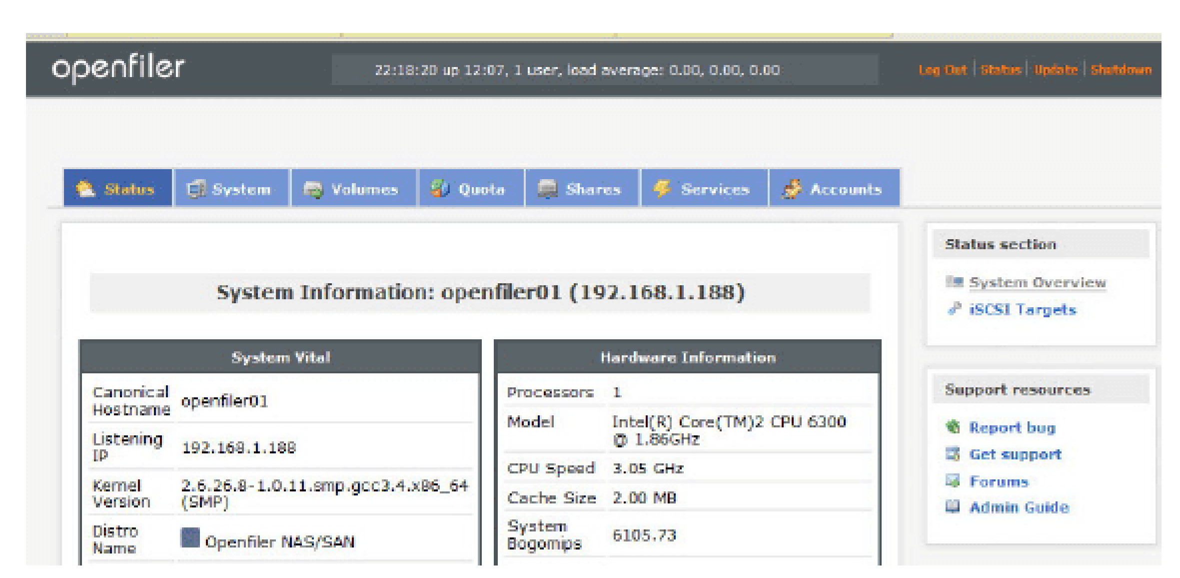
NAS 的底层当然也是不同的磁盘，也可以使用不同的界面，如 IDE、SATA、SCSI 等，但前提是这些磁盘在 NAS 运行的操作系统上必须能被辨识，并且可以安装对应的文件系统提供分享。在 NAS 的硬件底层，也可以使用硬件的 RAID 作为安全机制，并且可以在需要的时候，以 LUN 的方式供 NAS 本身的操作系统进行卷的管理。



▲ NAS 的外观和内部结构与 DAS、SAN 都没多大差别，主要还是传输界面

4. NAS 的配置

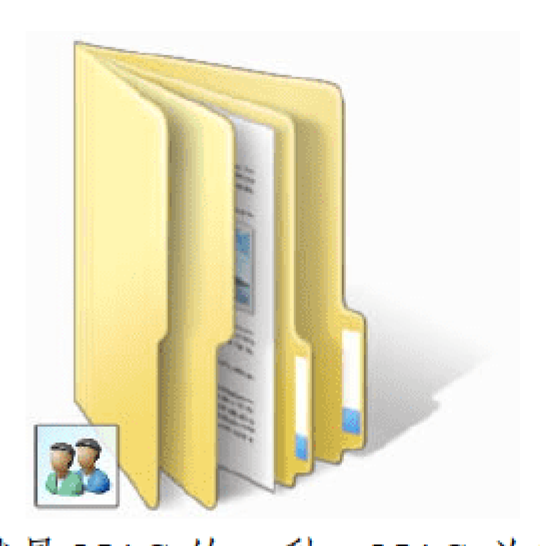
NAS 必须通过网络上的通信协议来配置，一般来说就是 HTTP 或是给定的客户端。当前大部分的 NAS 产品都是用 Linux 修改的操作系统作为配置工具，因此使用 HTTP 或是 HTTPS 协议连入给定的 IP，就会弹出修改的画面，并且可以进行基本的操作。



▲ NAS 一般是通过浏览器来配置的

5. NAS 的文件系统

由于 NAS 是以文件为主的共享机制，因此必须在磁盘之上有个文件系统。在 Linux 下使用的文件系统多为 NFS（Network File System），而在 Windows Server 下则使用 SMB（Server Message Block）或是 CIFS（Common Internet File System）。当然任何专业的 NAS 设备是不会局限于固化的文件系统的。



▲ 网络文件共享就是 NAS 的一种，NAS 必须以文件为基本单位

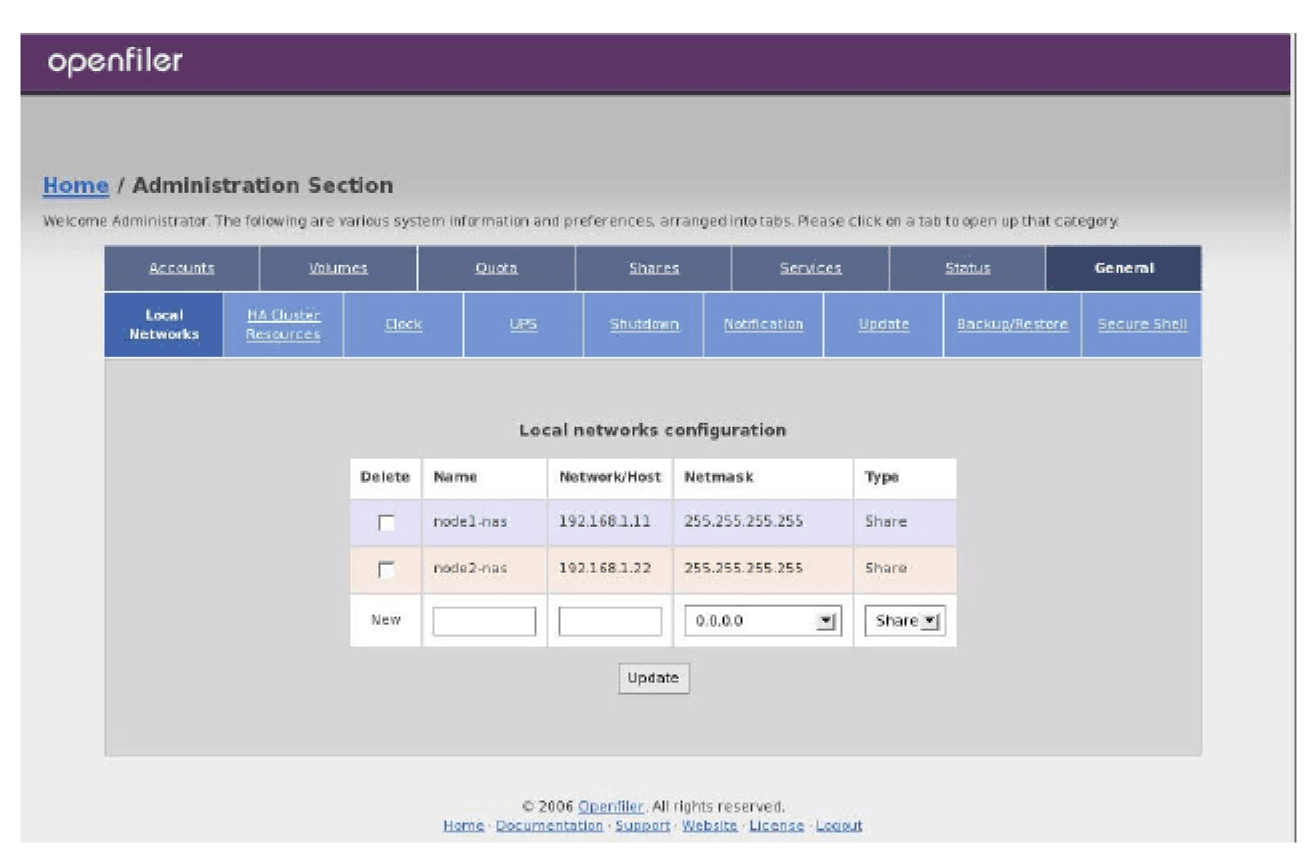
6. NAS 产品

总的来说，一台常规的 PC 安装上了 NAS 的操作系统之后，也可以成为一台 NAS，通常我们会将这台 NAS 称为软件的 NAS 系统。当然市面上也有专业的 NAS 系统，不一定使用 PC 的硬件架构，并且也有专业的冗余设备，如 RAID、多电源等。大部分的存储设备厂家都有 NAS 产品，而在宽带网络流行之后，用户大量下载数据，让 NAS 设备也走入家庭和个人，因此除了存储设备厂家之外，由于 NAS 是一个以网络为传输介质的产品，大部分的网络设备厂家也开始

生产个人/家庭用的 NAS 设备。



▲ 各大厂家都有 NAS 产品，算是最通用的独立存储设备



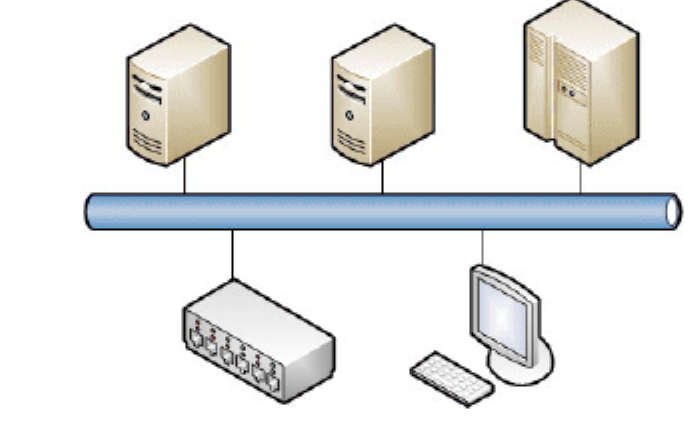
▲ 也可以自行在 PC 上安装 NAS 操作系统让这台 PC 成为一个 NAS

4.2.3 存储局域网络（Storage Area Network，SAN）

在企业的应用上，SAN 一直是最受系统管理人员青睐的存储产品。SAN 的功能强大，速度快、弹性大，并且具有以 Block Device 的低级处理方式，更可以让服务器访问逻辑磁盘（LUN）以及更低一层的卷（Volume Group），对虚拟机来说，SAN 的速度可贴近本地硬盘，在许多情况下，甚至可以在配合高速 HBA 光缆信道卡，取代本地硬盘，因此 SAN 可以说是虚拟机的最佳拍档。

1. SAN 的近亲 LAN

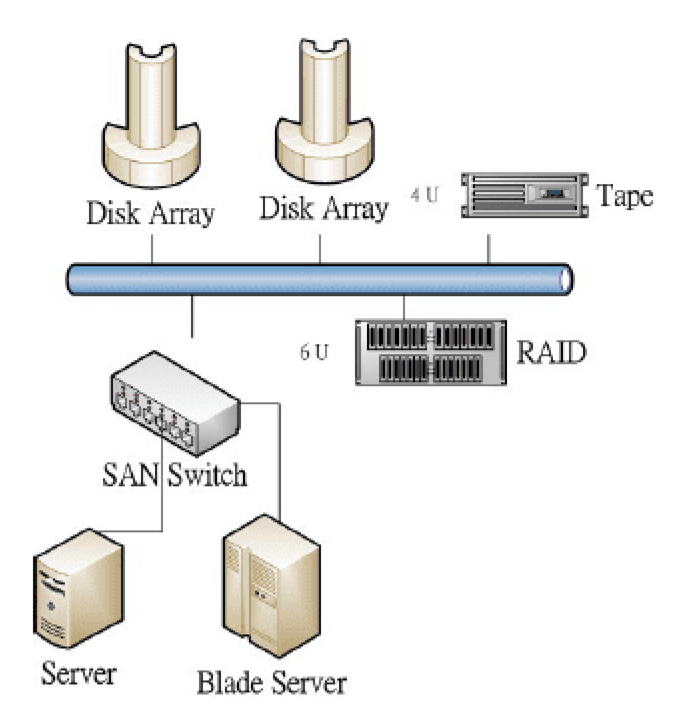
先不说 SAN 的定义，我们将另一个字义贴近的字 LAN 放在一起，你会发现只有第一个英文单词不同。LAN 就是 Local Area Network，这是大家都相当熟悉的词：局域网络。在同一个区域组成的网络。当我们将 Local 换成 Storage，我们得到的是“存储设备的网络”。事实上，SAN 的定义就是如此。



▲ 先来看看 LAN，就是一群计算机的组合

首先要确定的是，SAN 是一个完整的 OSI 七层网络架构，不是一种存储设备。这个网络是专门给主机连接存储设备用的，而这种新型态网络的任何部件，目的就是要让服务器能更便利地访问存储设备。

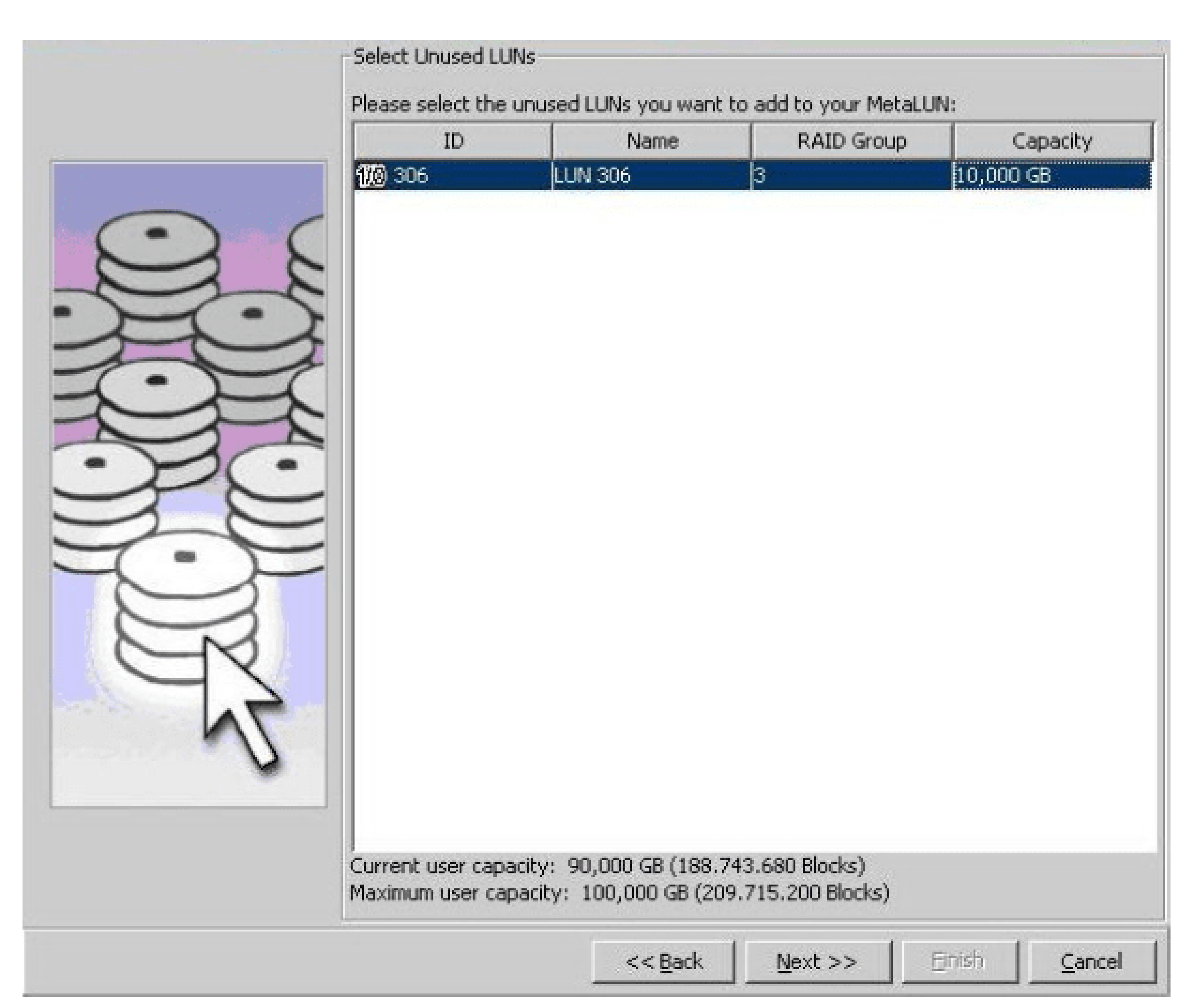
在拓扑上，SAN 和 LAN 可说是大同小异。举例来说，局域网络互相连接一般需要一个交换机，让连上网络的机器都通过交换机交换数据。而存储设备也需要一个交换机。在 LAN 上的计算机每一台都需要一个网卡，而在 SAN 中的每一台机器也需要一个“SAN 界面卡”连上 SAN 交换机。但 SAN 是用在存储上，因此不管是 HBA、交换机或是传输线路都必须在相当高的速度下运行，因为 SAN 和 LAN 所应付的数据量不是在同一级别的。SAN 在每秒钟所处理的单位通常是 GB 级的数据，而 LAN 则是数百 K 至数 MB 而已。



▲ SAN 就是将网卡换成 HBA 卡的一群由存储设备组成的集合

2. SAN 的定义

在理解了 SAN 的架构之后，我们就可以理解 SAN 的基本定义了。SAN 就是将不同的存储设备互相连接所形成的一个网络架构，这些存储设备包括了磁盘数组，磁带柜或是光盘柜。存储客户端（通常是服务器）也通过界面卡（通常为高速的 HBA，如光缆）连入这个网络中，存储网络对于访问它的计算机来说是完全透明的，访问的层级可以低到局块单元或单个物理磁盘，但大部分的访问单位还是以逻辑磁盘为主（LUN）。



▲ SAN 的划分一般以 LUN 为主

3. SAN 的通信协议

相对于 NAS 是以文件为主要访问单位，SAN 既然被视为一个等于本地硬盘的存储设备，当然必须通过一个能被服务器识别的信道，通过 SAN 交换机访问 SAN 存储设备。最常见到的 SAN 通信协议是光缆信道（Fibre Channel），或称为网状信道，因此我们常常看到“FC SAN”会连接在一起使用，当然除了光缆信道之外，SAN 也可以使用下列的通信协议来提交数据。

▶ **常见的 SAN 通信协议**

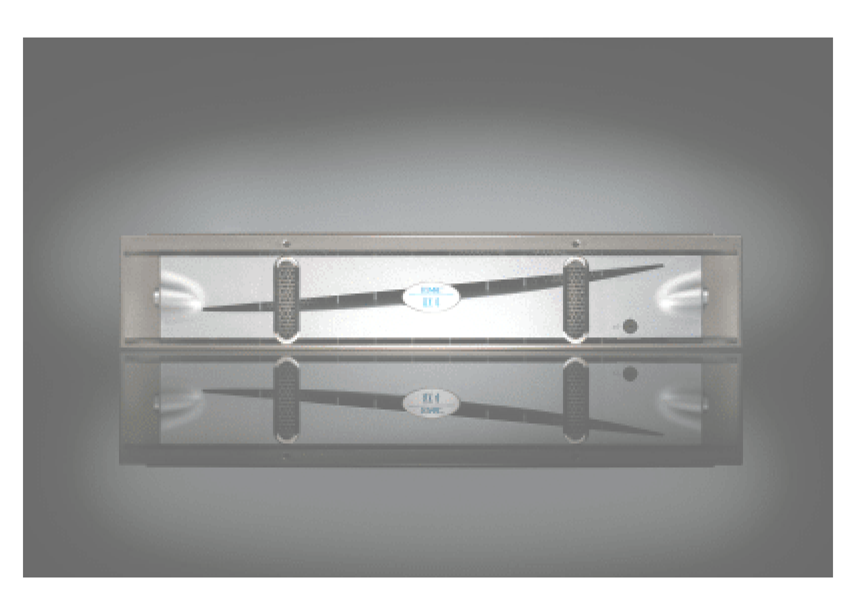
- Fibre Channel Protocol （FCP）：最常见的 SAN 通信协议，使用 SCSI over Fibre Channel。
- iSCSI：最常见的 SCSI over TCP/IP 协议，本书中的示例会大量使用。
- HyperSCSI：SCSI over Ethernet。
- ATA over Ethernet： ATA over Ethernet，较少在企业的产品中见到。

- FICON over FC：常见于大型机。
- FCoE：Fibre Channel over Ethernet。
- iSCSI Extensions for RDMA （iSER）：iSCSI over InfiniBand （IB）。
- iFCP 或称为 SANoIP：一种 FCP over IP。

> **注意　是 Fibre 不是 Fiber**
>
> FC 中的 F 通常代表的是 Fibre 而不是 Fiber。后者 Fiber 指的是使用光缆的设备，而 Fibre Channel 则是使用光缆信道的通信协议。但 Fibre Channel 的通信协议可以使用光缆之外的传输线材，Fibre Channel 另翻译成“网状信道”，取的是纤维交错的意思，和光缆完全无关。Fibre Channel 简称为 FC 协议、FC 网络或 FC 连接，是一种匹配 OSI 模型的网络协议，用在 SAN 之上，这是必须注意不可以弄混的。

4. SAN 的本体

SAN 网络中最常见的存储单位就是磁盘数组，其本体和 NAS 和一样，也是由不同界面的物理硬盘组成。但整机对外连接仍然是通过内置的 HBA 连上 SAN 交换机。在 SAN 网络中的磁盘数组中，多个硬盘也可以做成 RAID，其上也有 VG 和 LUN。但和 NAS 不同的是，在 SAN 中，连上交换机的不只一台存储设备，可能会有很多个或多种存储设备，因此不但可以做到单独物理磁盘等级的冗余，甚至可以做到整个机柜磁盘数组等级的冗余，在安全性上比 NAS 或 DAS 再高上很多等级。



▲ SAN 的本体也是标准的硬盘柜

5. SAN 的文件系统

SAN 最重要的特色就是其磁盘数组低级的访问，对访问 SAN 的计算机来说，可以将 SAN 上的一个逻辑分割区视为本地的一个硬盘。举例来说，在一个 50 TB 的 SAN 磁盘数组中，可以分割出一个 200 GB 的 LUN，然后另一台连接上这个 SAN 的 Windows Server 2008 服务器会认出这个 200 GB 的 LUN 为一个物理硬盘，我们可以进一步将这个 200 GB 的硬盘再划分成一个 50 GB 的引导磁盘 C 和 150 GB 的数据磁盘 D，并且在其上安装 NTFS 和 FAT32 文件系统。当这台 Windows 2008 蓝屏时，我们可以拿另一台一模一样的服务器直接接上这个 LUN 引导，就可以完全省去重新安装的麻烦，这是 NAS 和 DAS 完全无法抗衡的优点。

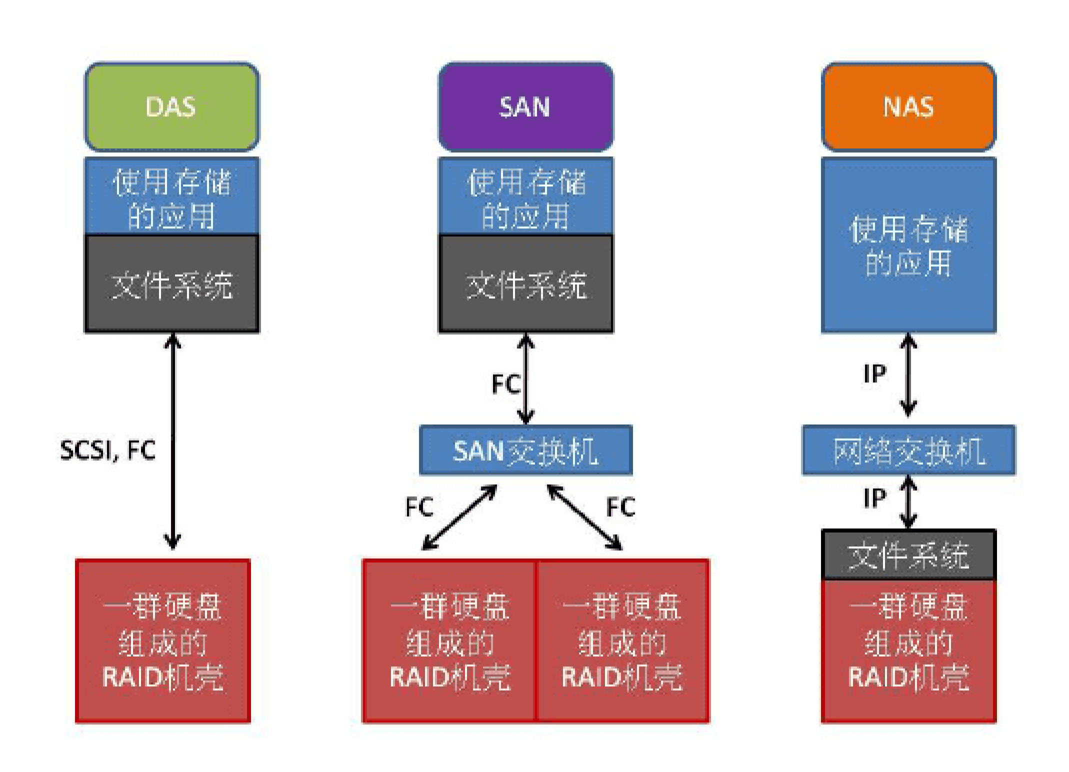
6. SAN 和 vSphere

vSphere 的 ESX/ESXi 都可直接对应到 SAN 的 LUN 上，如果有需要的话，更可以使用 Raw Device Mapping（RDM）将整个 LUN 对应到一个虚拟机上，通常发生在 P2V 转换的某个会话。不过在一般的使用上，我们会使用 SAN 上的一个 LUN 对应到多个 ESX 组成的 VI 架构来放置所有的虚拟机文件，当某一个 ESX Server 蓝屏时，可以立马激活另一个 ESX Server 来可读这个 LUN 上的所有虚拟机，而达成 HA 或 VMotion 的目的。虽然在 NAS 和 NFS 上也可以这么做，但 SAN 的速度和针对硬件直接访问的弹性，是和虚拟机搭配最完美的外置式存储设备。

7. 廉价的 SAN 方案：iSCS

在企业中最常见的是 FC SAN，但 FC SAN 不管是 SAN 存储设备或是访问 SAN 的服务器都

需要昂贵的 HBA 卡，在测试环境或中小型企业的应用上，HBA 的成本和预算门槛，并不一定足够架设起 FC SAN 的整体环境。在互联网流行之后，TCP/IP 成为通信的主流，而将 SCSI over IP 这个通信协议善加运用之后，让 IP 网络也能成为 SCSI 的提交信道，这是新兴 SAN 的通信协议，称为 iSCSI，下一小节会有 iSCSI 的详细说明。



▲ 图为 DAS、SAN、NAS 的对比表

4.3 vSphere 最佳性价比伙伴：iSCSI

在 vSphere 环境中使用独立的存储设备是首要条件，对于较大的企业来说，FC SAN 应该早就是必备的设备了。但针对中小企业来说，在没有经过完整的虚拟化评估之前，直接选购存储设备是不匹配成本效益的。

然而 IT 人员如果没有独立的存储设备又无法进行虚拟化的测试和评估，让系统管理人员在进行虚拟化评估时陷入两难。还好当前市面上有许多廉价甚至是免费的 NAS 软件套件，允许用户将普通的计算机摇身一变成为 NAS，甚至有些产品提供了 iSCSI 连接选项，让具备 iSCSI 连接功能的操作系统（如 Windows Server 2008 或 XP）能直接访问其上的 LUN，因此使用现有的 IP 网络，再加上 SCSI 命令，iSCSI 立即能在不添加成本的情况下创建 SAN 环境。

4.3.1 什么是 iSCSI

在虚拟化越来越流行之后，存储设备也成为一个新兴的主流趋势。然而在提到存储设备时，是不能不提到 SCSI 这个界面的。SCSI 一直是存储设备最重要的界面之一，大部分的 SAN 或 NAS 专业产品都是使用 SCSI 硬盘，那 iSCSI 和 SCSI 之间的关系为何呢？我们就先来看看。

1. IP 网络的兴起

随着 IP 网络的速度越来越快，IP 网络也在 Internet 流行之后成为网络的主流，越来越多的数据都可以通过 IP 网络在局域网或是广域网之间提交。有鉴于 SAN 网络设备及界面卡的成本一直居高不下，成为企业购买独立存储设备的最大障碍，大部分的存储设备制造商也绞尽脑汁，希望在存储设备的存放物理之外，能帮用户省下大量的成本，但并发又不能牺牲提交数据的稳定性和速度，因此利用现存的 IP 网络，是最省钱，也是最方便的解决方案之一。

> **注意**
> SCSI 是命令集而非硬件
> 常用服务器等级的用户一定会觉得很奇怪，SCSI 不是硬件的扩展卡吗？怎么可以在“网卡”上使用？事实上 SCSI 是一个命令集，用来连接设备之间，从其名称 Small Computer System Interface 就可以得知，是计算机系统之间的界面，因此只要能将 SCSI 命令架设在任何物理的连接线路上，就可以让连接线两端的设备利用 SCSI 命令来提交数据，甚至是最底层的分割区数据。

2. iSCSI 的弹出

iSCSI 全称为 internet Small Computer System Interface，是由 IEETF 开发的网络存储标准，从其名称就可以知道，是 IP 通信协议和 SCSI 存储设备的组合，换句话说，就是通过在 IP 网上提交 SCSI 命令和数据。iSCSI 由于使用了 IP 网络，因此可以在网际之间传递，更促进了数据的远距离管理，可说是提速 SAN 市场快捷发展的关键因素之一。由于使用了 IP 网络，iSCSI 能够在 LAN、WAN 甚至 Internet 上进行数据提交，使得数据的存储不再受地域的限制。



▲ 当前主流的 OS 都支持 iSCSI

3. iSCSI 和 Ethernet 的结合

当前大部分的网络架构都是在低层使用 Ethernet，上层使用 IP 通信协议，因此 iSCSI 必然会使用 Ethernet 来提交数据。用户可使用标准的 GB 级 Ethernet 网络通信协议，通过 Cat5 网络线和各种 GB 级的交换机来提交，如此一来，计算机上的 GB 网卡不但具有网络提交功能，更成为一个访问外部 SCSI 设备的 SCSI 卡了。

4.3.2 iSCSI 的连接方式

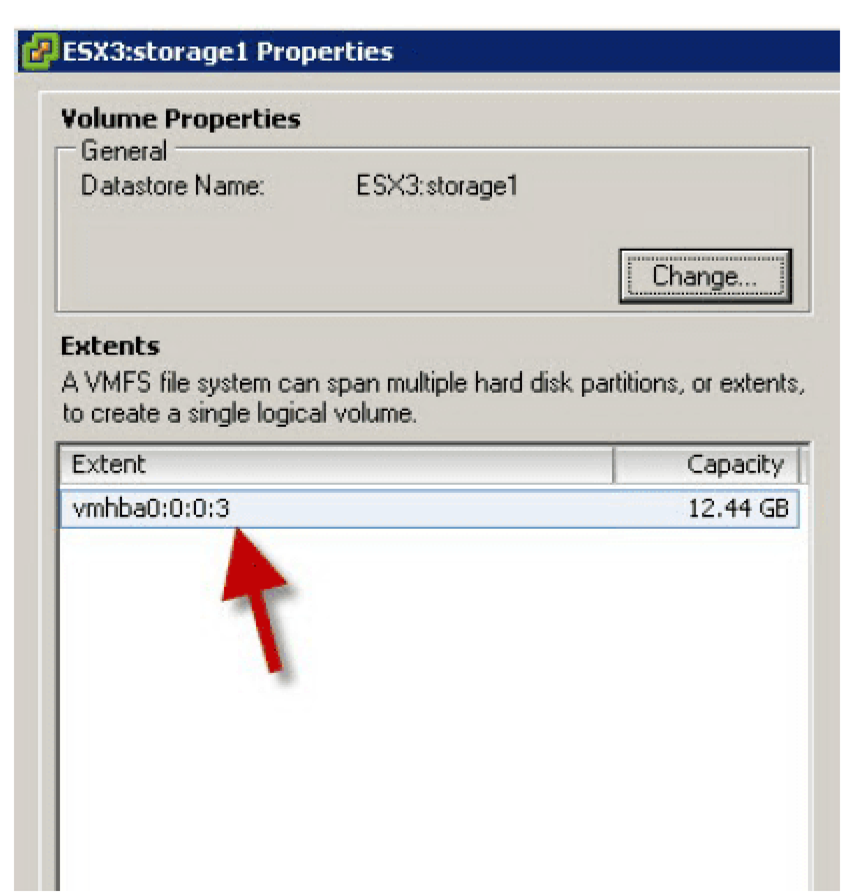
利用 IP 来运行 SCSI 命令这个思路十分实用，但我们如何创建一个 iSCSI 网络存储系统呢？iSCSI 设备的主机界面一般都是使用 IP 通信协议，直接与 Ethernet 网络交换机或 iSCSI 交换机连接，形成一个类似 SAN 的架构。而根据主机端的 HBA 卡、网络交换机的不同，iSCSI 设备与主机之间有三种连接方式，但基本上，iSCSI 的连接都是由需要的主机发起 Initiator 要求，而存储设备本身拥有一个 iSCSI 的 Target，在接收到需求后再创建起连接。

1. Ethernet 网卡及 Initiator 软件

这是最省钱的一种做法。在 LAN 中的所有计算机使用标准的 Ethernet 网卡，通过 Ethernet 网线直接与交换机连接，iSCSI 设备也是通过 Ethernet 网线连接到 Ethernet 交换机上，甚至直接连接到主机的 Ethernet 网卡上。

想要访问 iSCSI 设备的计算机，则需要安装 Initiator 软件来连接存储设备。在安装 Initiator

软件后，Initiator 软件可以将 Ethernet 网卡虚拟成 iSCSI 卡，接收和发送 iSCSI 数据，实现主机和 iSCSI 存储设备之间的 iSCSI 协议和 TCP/IP 协议传输功能。由于使用了普通的网卡和交换机，因此不需要购买新的硬设备，是最节省成本的解决方案，但效能上并无法满足访问频繁的大型环境。当前所有的主流操作系统都提到了免费的 iSCSI Initiator，如 XP、Windows Server 等，而 VMware 的 ESX Server 也提供了内置 iSCSI 连接软件。



▲ 在 ESX 下也可以用 iSCSI 连接，在稍后的章节会有说明

注意

使用网卡引导的 iSCSI 方式

许多网卡支持 PXE BOOT 引导，也就是从网卡连接到远程的硬盘引导。当前有 SAN BOOT 专业条目支持从 PXE 连接到 iSCSI Target 上的 LUN 引导，但 ESX/ESXi 并不支持，有兴趣的读者可以到 http://etherboot.org/wiki/看看。



▲ 可以 PXE Boot 的方式用 iSCSI 引导

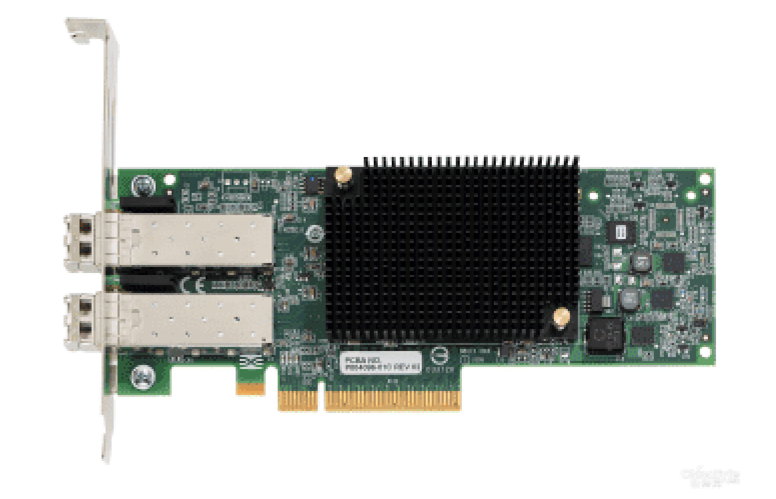
2. 硬件 TOE 网卡/Initiator 软件

使用纯 Ethernet 和软件的方式由于都需要主机 CPU 来进行运算，数据传输的速度直接受到主机当前运行状态和可用资源的影响和限制，因此一般无法提供高带宽和高 I/O 速度。因此将运行 iSCSI 命令的任务交到硬件上的界面卡可以加快传输的速度。在不需要占用一个 PCI 插槽的思路下，将硬件处理放上 Ethernet 网卡的产品即在高度的需求下弹出了。在这种解决方案中，将具有 TOE（TCP Offload Engine）功能的芯片放上 Ethernet 网卡，就可以将网络数据流量的处理任

务全部转到网卡上的集成硬件中进行，计算机只承担TCP/IP 总控信息的处理任务即可。

3. 使用专属的 iSCSI HBA 界面卡

在主机上安装专业的 iSCSI HBA 界面卡，从而实现主机与交换机之间、主机与存储设备的完整 iSCSI 提交。但由于费用和使用 FC SAN 并无太大差别，因此不适合于在乎成本的中小型企业。



▲ 专业的 iSCSI 卡

4.3.3 最基本的 iSCSI 架构：以 PC 为主的解决方案

基本上 iSCSI 是利用 IP 网络来提交 SCSI 命令，因此除了 IP 网络之外，也需要落实 SCSI 命令的实现，不论是使用软件或是硬件的手段。由于 iSCSI 已经是一个业界的通用标准，并且大量应用在 FC SAN 上，因此各家厂家都有自身的 iSCSI 实现解决方案，但在以节省成本为主的中小企业，我们还是以 X86 架构为主。

▶ 以 X86 为主的 iSCSI 架构

X86 架构就是存储设备创建在 PC 服务器的基础上，找一台能安装多个磁盘，并且支持 RAID、热插拔功能的计算机直接作为 iSCSI 的存储设备，再选择一个相对成熟稳定的 iSCSI Target 软件，将 iSCSI Target 软件安装在 PC 服务器上，使普通的 PC 服务器转变成一台 iSCSI 存储设备，并通过 PC 服务器的 Ethernet 网卡对外提供 iSCSI 数据传输通信协议。

在 PC 架构的 iSCSI 存储设备上，所有的 RAID、LUN、iSCSI、TCP/IP 运算都是以软件方式落实，对 CPU 和内存的要求较高，而 iSCSI 存储设备的性能也很容易被 X86 服务器当前负担所影响。

由于 PC 架构 iSCSI 存储设备的研发、生产、安装使用都比较简单，硬软件成本相对较低，因此市场上常见的基于 PC 架构的 iSCSI 设备的价格都比较低，在一些对性能稳定性要求较低的系统中具有较大的价格优势，本书所使用的示例，也都是这种架构。

Enable/Disable services

Service Name	Status	Modification
SMB/CIFS	Disabled	Enable
NFSv3	Enabled	Disable
HTTP / WebDAV	Disabled	Enable
FTP	Disabled	Enable
iSCSI target	Enabled	Disable
Rsync	Disabled	Enable
UPS	Disabled	Enable
LDAP	Disabled	Enable

▲ 使用 OpenFiler NAS 系统，也支持 iSCSI

4.3.4 虚拟机上最常见的 iSCSI 使用

在理解了 iSCSI 之后，我们在这一小节就来看看，在虚拟机中如何利用 iSCSI 作为存储界面，用来访问存储设备上的数据。vSphere 访问 iSCSI 的方式有很多种，大多数是以直接对应到 iSCSI 上的逻辑单位，即以一个虚拟的硬盘为主，我们称为 LUN。对 vSphere 来说，LUN 是最基本的访问单位，可以直接安装文件系统，这对虚拟机的效能或是管理都较有优势。下面就是几个最常见的连接演示。

1. 每个主机有自身对应的 LUN

这种方式是在 iSCSI 设备上创建多个 LUN，不同的 LUN 给不同的主机使用，每个主机分别管理和访问自身的 LUN，好像将多个主机的本地磁盘集中放置在一个网络设备中，各主机之间仅实现硬设备层的共享。这种方式最常见于独立的 ESX/ESXi。

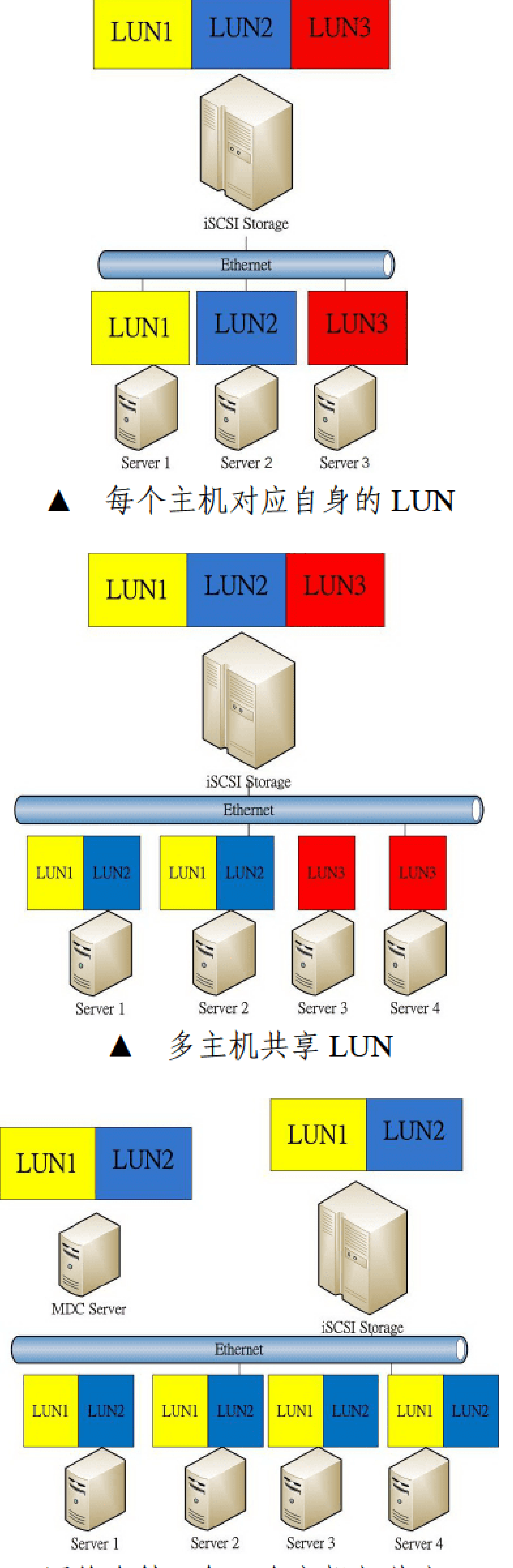
2. 多主机备援的 LUN 共享

在使用 Cluster 时，数据库或 Web 的备援机器在激活时，往往需要访问激活之前主机访问的数据，因此需要将 LUN 对应到不同的主机上，两台成对的服务器之间需要管理和使用相同的 LUN。在这种架构下，主机之间可以通过 MSCS 或 RAC 等软件实现 Cluster 功能。这种方式最常见于使用在 HA、VMotion 等场合。

3. 网络存储

有时主机之间只需要有共享的文件夹，并不需要备援或是进行较高级的虚拟机功能时，只要保证 iSCSI 上的 LUN 能被各主机可读即可。这时我们就可以使用 iSCSI 上的 LUN 对应来访问兼容的数据。这种情况通常在 NAS 中弹出。

为了保证同一个文件系统可以并发被多台主机使用，且不会因为并发访问同一个文件而引起文件系统管理冲突，通常会在所有需要文件系统共享的主机上安装 SAN 网络存储共享管理软件，并设置一台或多台主机为管理软件的服务器端。该服务器负责管理主机之间的元数据交换，称为 MetaData Controller（MDC），而 MDC 正是调解不同访问时冲突的最好工具，在大部分的专业 SAN/NAS 设备都会使用 MDC。



▲ 每个主机对应自身的 LUN

▲ 多主机共享 LUN

▲ 网络存储，每一个主机都共享 LUN

结 语

理解不同存储设备对 vSphere 来说是最重要的，vSphere 支持 NFS、DAS、NAS、SAN，而最新的 iSCSI 更是成本和性能的最佳组合，在配合了 LUN 的使用之后，更可以实现将 VM 直接对应到存储设备上，这对整个企业 IT 环境来说，更增添了确定性以及部署的弹性。

第 5 章
vSphere 最重要基础：独立存储设备的软件实践

关键词：

- RAID
- RAID 的原理
- RAID 的种类
- 不同种类的 RAID 在安全性效能上有何不同
- Volume
- Volume Group
- LUN
- 文件系统
- LVM 和 RAID 的异同
- 创建 NAS 设备的 RAID1/5

不管是哪一种存储设备，现在都很少使用单个的硬盘来操作，主要还是为了安全及性能。在 vSphere 下，支持各式各样的存储设备，如 DAS、NAS、FC SAN 及 iSCSI。当然网络最流行的 NFS 也没问题。这些存储设备在真正能给 vSphere 使用之前，必须经过空间的安排，而当前所有存储设备的空间安排，首先想到的就是 RAID。当然在 RAID 之下，就是生成卷的管理了，通常都是通过 Volume 的管理。在这一章，我们就来看看 Volume 的管理，并且以具体的产品来进行配置演示。

5.1 什么是 RAID

不管是安装 ESX/ESXi 或是使用独立存储设备，我们都必须将硬盘规划成一个安全且方便使用的环境，而 RAID 就是当前最广泛使用，也是成本最低的高可用性的硬盘空间管理，我们在这一小节，就先来理解一下什么是 RAID。



▲ RAID 的目的是将多台硬盘组合成一个大的存储池

5.1.1 RAID 的原理

RAID 全称为“廉价重复磁盘数组”Redundant Arrays of Inexpensive Disks，意即使用一般商用产品的多台硬盘达到高可用性或是提高性能的目的。而根据使用的硬盘数、目的以及容许错误的硬盘数量，也有不同的组合。

1. RAID 的由来

RAID 顾名思义，就是用多台硬盘组成一个大的存储池，由于在 RAID 生成的年代（约 1980 年），专用的高可用性硬盘通常为高价产品，因此发明了 RAID，以期能使用市售产品，以更低的成本达到同样的目的。

2. RAID 的原理

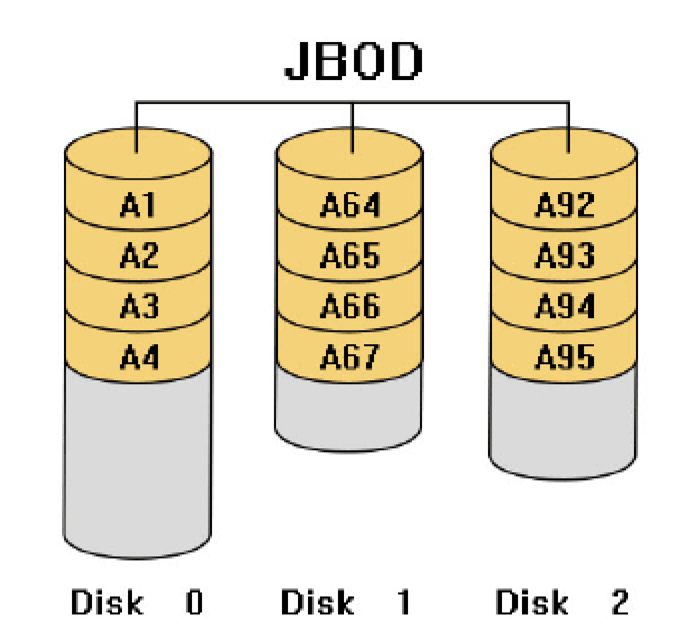
RAID 的原理相当简单，就是利用 XOR 的计算方式来推算丢失的数据，举例来说，A XOR B = C 是一个基本的运算，在 XOR 的原理之下，A XOR C =B，或是 B XOR C=A 也是成立的。因此在 A、B、C 中，有任何一个数据丢失了，都可以利用 XOR 将剩下的数据计算，因而复原丢失的数据。如果我们假定将 A、B、C 三个数据放在三个不同的硬盘中，那么当其中一台硬盘坏掉时，就可以由剩下的两台硬盘计算出坏掉硬盘中的数据，从而恢复完整的数据。

5.1.2 RAID 的种类

根据不同的排列组合，RAID 也有不同的硬盘组合方式，下面就是最常用的几种 RAID 形式。

1. JBOD（Just a Bunch Of Disks）

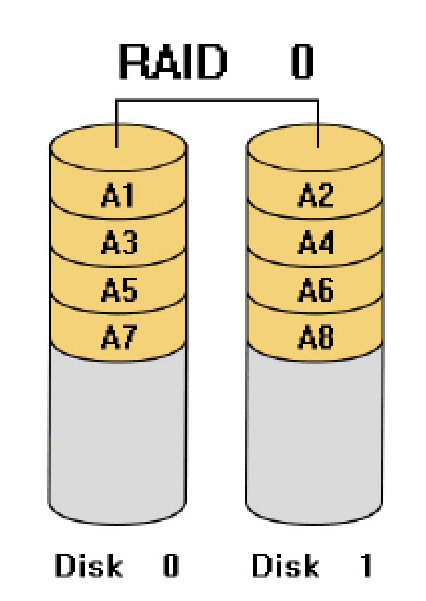
JBOD 不算是 RAID，因为完全没有安全性，只是把一大堆硬盘并成一个大硬盘的工具而已。JBOD 的存储是将所有数据先存在第一个硬盘上，当第一个硬盘满了之后就存上第二个，但由于第一个硬盘有可能是硬盘的分区表，因此如果第一个硬盘坏了，后面的数据就读不出来了。JBOD 没有任何性能和安全上的提高，就是为了方便而已。



▲ JBOD 通常只是将许多硬盘组合起来，没有任何安全及性能上的提高

2. RAID 0

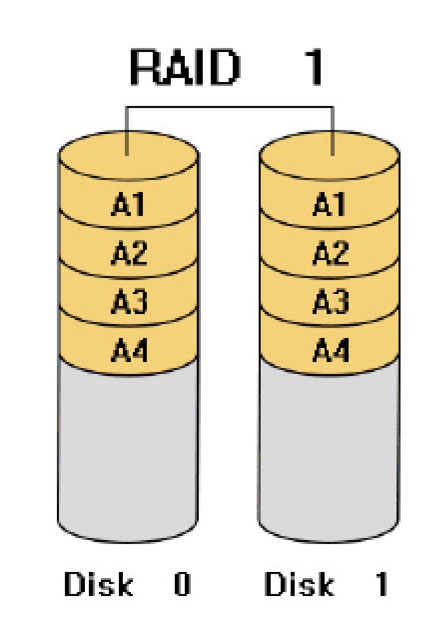
- 原理：RAID 0 又称为 Strip，是将所有 N 多硬盘加总的方式。当压入数据时，会将数据拆成 N 份，压入不同的硬盘。
- 参与硬盘：至少需要两颗，容量为（最小容量硬盘×n），三颗 500GB 的硬盘，组成 RAID 0 时，将会生成一个 1500GB 的大硬盘。
- 安全性：RAID 0 不具备任何安全模式，因此当任何一个硬盘故障时，整个数据就不完整，就无法可读了
- 性能：所有数据的读写都是 N 倍速度，可以并发读写 N 多硬盘。
- 使用场合：需要速度，但不需要数据安全，如暂存盘或是 Swap。
- vSphere 使用：所有 VM 的 Swapfile 存放处，可以使用 RAID 0。



▲ RAID 0 有性能上的提高，但一个硬盘坏了就完了

3. RAID 1

- 原理：RAID 1 又称为 Mirror，是将 2 个硬盘使用镜像对应的方式。当压入数据时，会将数据落实镜像克隆，并发压入两个硬盘中。
- 参与硬盘：需要两颗，容量为（最小容量硬盘）。2 颗 1TB 的硬盘，组成 RAID 1 时，将会生成一个 1TB 的大硬盘。
- 安全性：RAID 1 具备镜像模式，因此当任何一个硬盘故障时，另一个对应的硬盘都有完整的数据在，可以继续使用，允许坏掉 1 个硬盘。
- 性能：RAID 1 只有可读时是两倍速，因为可以从不同的硬盘读相同的数据，但可写时没有速度的添加。
- 使用场合：算是最完整的安全组合，随时都有两个完整数据的硬盘。
- vSphere 使用：安装 ESX/ESXi 的本地，以及独立存储设备。



▲ RAID 1 在安全和性能上都有提高，就是空间有点浪费

4. RAID 10

- 原理：是 RAID 0 和 RAID 1 的合体，可以说是将两个 RAID 1 的 RAID 再组成一个 RAID 0 的概念。
- 参与硬盘：最少需要 4 颗，容量为（最小硬盘容量）×n/2，4 颗 1TB 的硬盘，组成 RAID 10 时，将会生成一个 2TB 的大硬盘。
- 安全性：RAID 10 具备镜像模式，因此当任何一个硬盘故障时，另一个对应的硬盘都有完整的数据在，可以继续使用。
- 性能：RAID 10 可看做 RAID 1 组合成的 RAID 0，因此可读是 N 倍，可写是 n/2 倍。

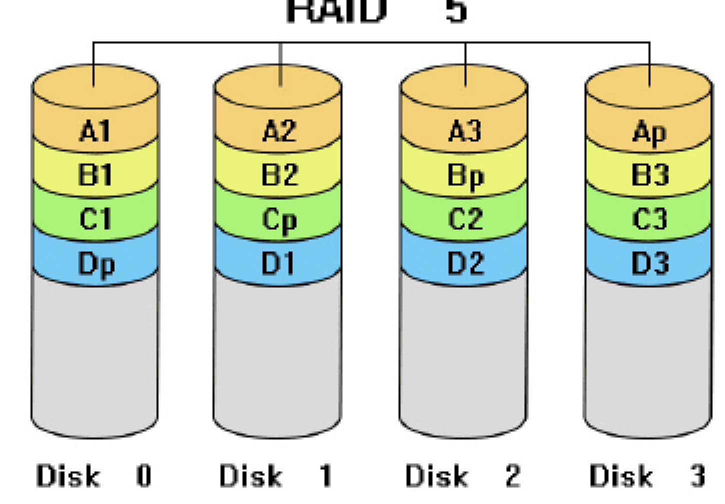
- 使用场合：并发需要高效能及高安全性，如图像编辑。
- vSphere 使用：通常使用在多台硬盘的独立存储设备。

5. RAID 01

- 原理：是 RAID 1 和 RAID 0 的合体，可以说是将两个 RAID 0 的 RAID 再组成一个 RAID 1 的概念。
- 参与硬盘：最少需要 4 颗，容量为（最小硬盘容量）×n/2，4 颗 1TB 的硬盘，组成 RAID 10 时，将会生成一个 2TB 的大硬盘。
- 安全性：RAID 01 具备镜像模式，因此当任何一个硬盘故障时，另一个对应的硬盘都有完整的数据在，可以继续使用。
- 性能：RAID 01 是性能最好的 RAID 组合。
- 使用场合：性能为主的场合，兼具安全性。
- vSphere 使用：通常使用在多台硬盘的独立存储设备。

6. RAID 5

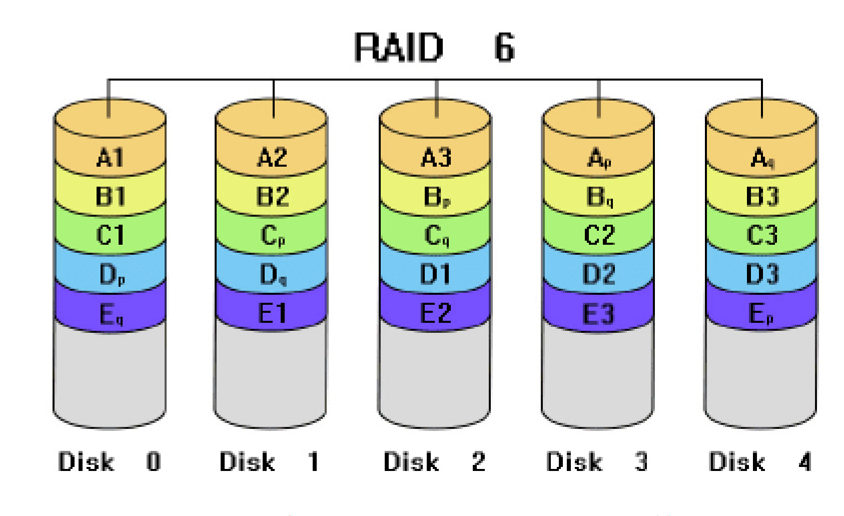
- 原理：使用多台硬盘，并且使用 XOR 生成校验数据，将校验数据分散在多个硬盘上。
- 参与硬盘：最少需要 3 颗，容量为（最小硬盘容量）×（n −1），3 颗 1TB 的硬盘，组成 RAID 5 时，将会生成一个 2TB 的大硬盘。
- 安全性：RAID 5 具备校验模式，因此当任何一个硬盘故障时，剩下的硬盘可以经过 XOR 计算获取丢失的数据，可以继续使用，但速度会慢下来，因为需要运算。需要立即将坏的硬盘补丁，允许坏掉一个硬盘。
- 性能：RAID 5 需要进行校验，因此视 RAID 卡上芯片的运算能力而定。
- 使用场合：空间、性能和安全的平衡，如独立存储。
- vSphere 使用：安装 ESX/ESXi 的主机（多硬盘）或是独立存储设备。



▲ 是最常用的 RAID 组合

7. RAID 6

- 原理：使用多台硬盘，并且使用 XOR 生成两组校验数据，将两组校验数据分散在多个硬盘上。
- 参与硬盘：最少需要 4 颗，容量为（最小硬盘容量）×（n −2），4 颗 1TB 的硬盘，组成 RAID 5 时，将会生成一个 2TB 的大硬盘。
- 安全性：RAID 6 具备双重校验模式，因此当任何两个硬盘故障时，剩下的硬盘可以经过 XOR 计算获取丢失的数据，可以继续使用，但速度会慢下来，因为需要运算。需要立即将坏的硬盘补丁，允许坏掉两颗硬盘。
- 性能：RAID 6 需要进行校验，因此视 RAID 卡上芯片的运算能力而定。
- 使用场合：着重安全、空间，较不着重性能，如独立存储。



▲ 在安全较被重视场合，可以使用 RAID 6

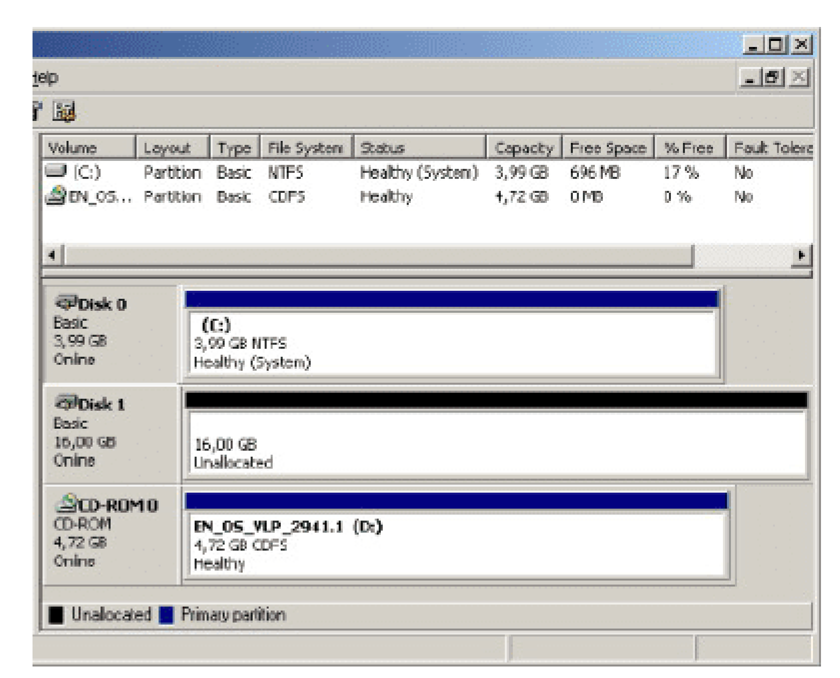
- vSphere 使用：着重安全的安装 ESX/ESXi 的主机（多硬盘）或是独立存储设备。

5.1.3 RAID 的实践观念

RAID 是一个观念而非一个产品，因此只要是能将硬盘组合的实践都可以叫做 RAID。我们就以不同的角度来看看。

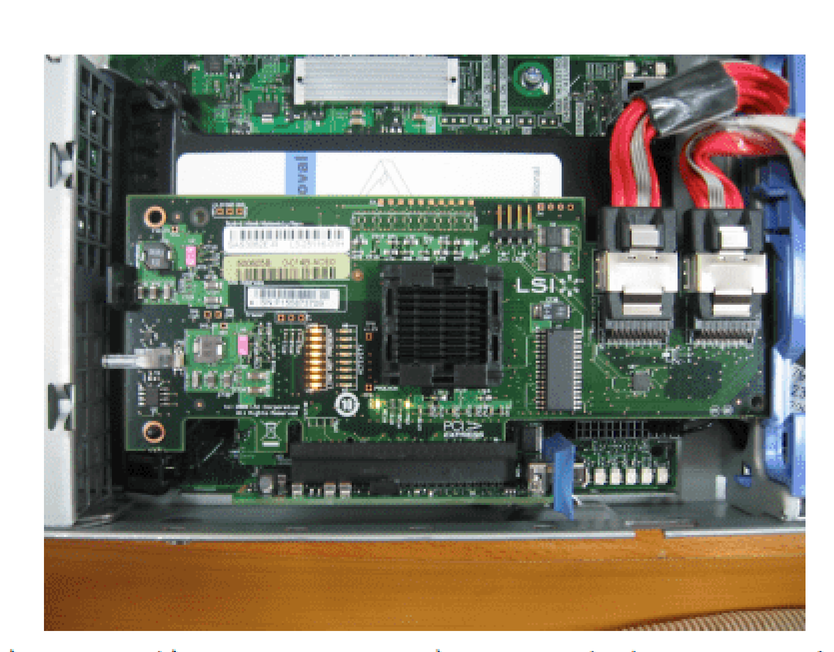
1. RAID 的软硬件区别

由于 RAID 牵涉到 XOR 的计算，而 XOR 是比特等级的运算，因此可以使用主系统的 CPU，或是专门处理 XOR 的芯片组来落实。一般来说，使用计算机上 CPU 来计算 XOR 的 RAID 方式通常称之为软件 RAID，这种 RAID 可以由操作系统来实践，也可以由 BIOS 提供的界面来实践，但通常速度都较慢，需要开销主系统的 CPU 资源。



▲ 在操作系统下也可以做 RAID，但具体意义不大

专业的服务器或是存储设备通常会使用专属的芯片来进行 XOR 的运算，这种芯片通常也做成界面卡，称为 RAID 卡。RAID 卡使用自身的芯片计算，不开销主系统的 CPU，也会提供一个 BIOS 来落实 RAID 的创建。当前有 SATA 和 SCSI 的 RAID 卡，用来连接不同的硬盘，而 SAS 界面的 RAID 卡是当前兼容性最高的，不但可以连接 SATA 硬盘，也可以连接 SAS 硬盘，让 RAID 的创建更加有弹性。

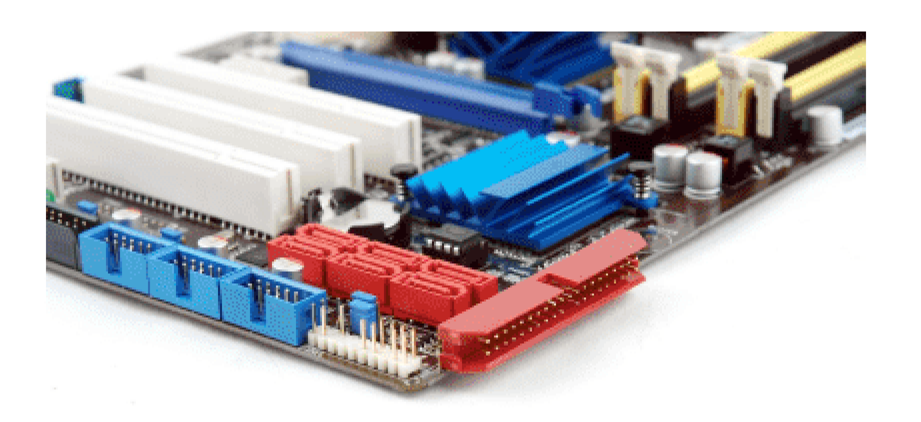


▲ 图为 IBM 的 LSI RAID 卡，可以让 SATA 和 SAS 的硬盘共存

注意

真假硬件 RAID

在 SATA 流行之后，Intel 的 BIOS 开始推出 RAID 功能，而有些主板上内置的芯片组也宣称有 RAID 功能，不管是 SATA 或是 SCSI 的 RAID。但这些 RAID 只具有创建以及组合硬盘的功能，并不具真正计算的功能，还是需要使用主系统的 CPU 来进行 XOR 的计算，因此并无法给 ESX/ESXi 使用。在真正使用之前，还是要到 vSphere 的 HCL 中检视确定支持，才能使用真正的 RAID 功能。



▲ 主板芯片内置的 RAID 通常被称为 FakeRAID

2. RAID 硬盘的热副本（Hot Spare）

在具备安全性的 RAID 组合上（1/5/6），当硬盘弹出故障时，我们必须将故障的硬盘更换，并且重建（Rebuild），将数据还原到更换后常规的硬盘上。大部分的高级 RAID 卡允许我们事先就将硬盘准备好，当 RAID 真正弹出问题时，能自动将新的硬盘换上进行重建，而不需要手动更换，这一类的功能称为热副本（Hot Spare），我们在最后一小节会有热副本的演示。

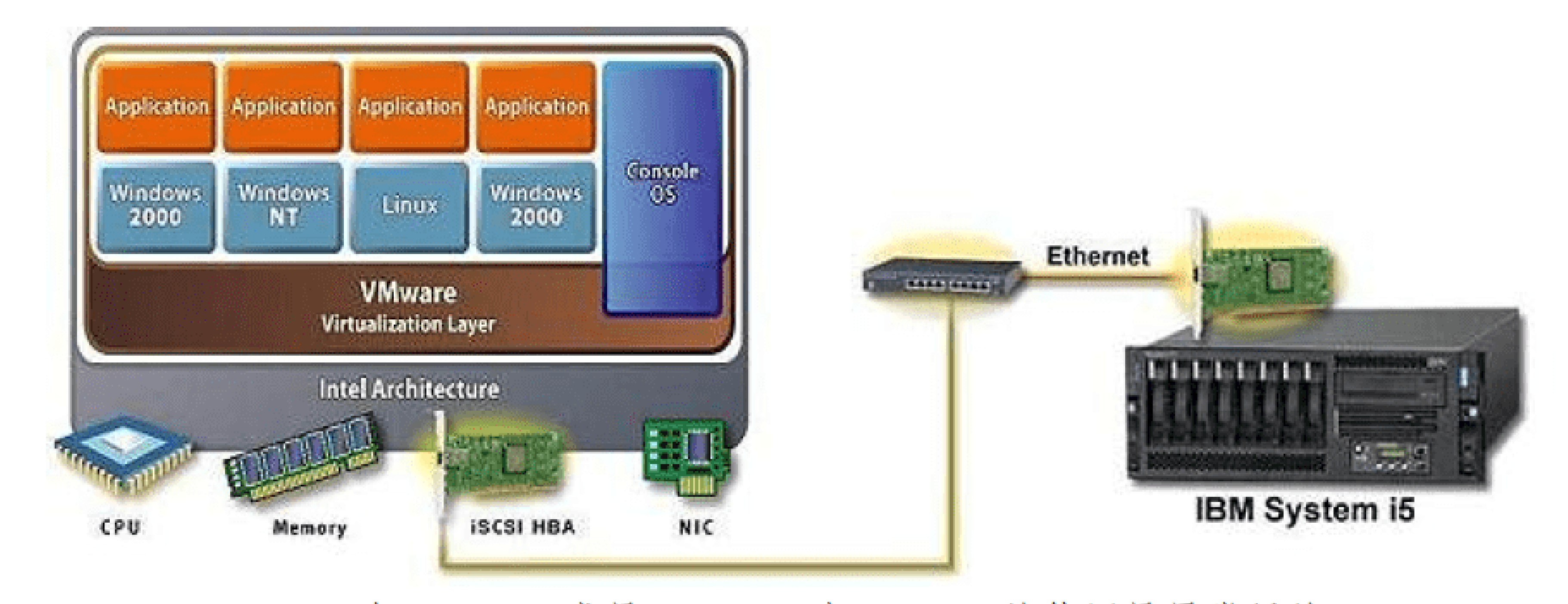
5.2 理解独立存储设备的操作方式

在上一章我们介绍内部和外部存储，vSphere 是并发支持两种存储方式的。事实上，在 vSphere 环境中，我们将存放在硬盘中的数据分成 Hypervisor、VM，以及其他数据（如实用程序或文档）。而在 vSphere 环境中，这三种数据都可以放在内部或外部存储空间中，一般来说，Hypervisor 和 VM 是我们最重视的数据，而其他数据一定是放在外部存储空间，我们就来看看几个对比典型的 vSphere 存储拓朴。

5.2.1 Hypervisor 的置放空间

Hypervisor 就是 ESX/ESXi，也就是虚拟机操作系统。我们可以将 Hypervisor 想成 Linux 或是 Windows 操作系统，在服务器上必须能安装这个操作系统才能引导。但在配合了外部的存储空间之后，Hypervisor 也可以安装在非本地的硬盘上。

我们在第 4 章弄清楚了 DAS、NAS 和 SAN 的区别，不管是 Hyper-V 或是 VMware 等虚拟机产品，都支持上述不同的存储设备，然而这三种设备最大的区别就是访问的方式。DAS 被视为本地连接设备，因此在安装时，只要能被操作系统或 Hypervisor 认出即可直接使用，NAS 则是以文件为基础的传输方式，要访问 NAS 的机器必须上面已经安装好操作系统了。而 SAN 则是综合了两者的优点，不但可以将本地的操作系统安装在 SAN 上，也可以用网络连接的方式来访问其上的文件。

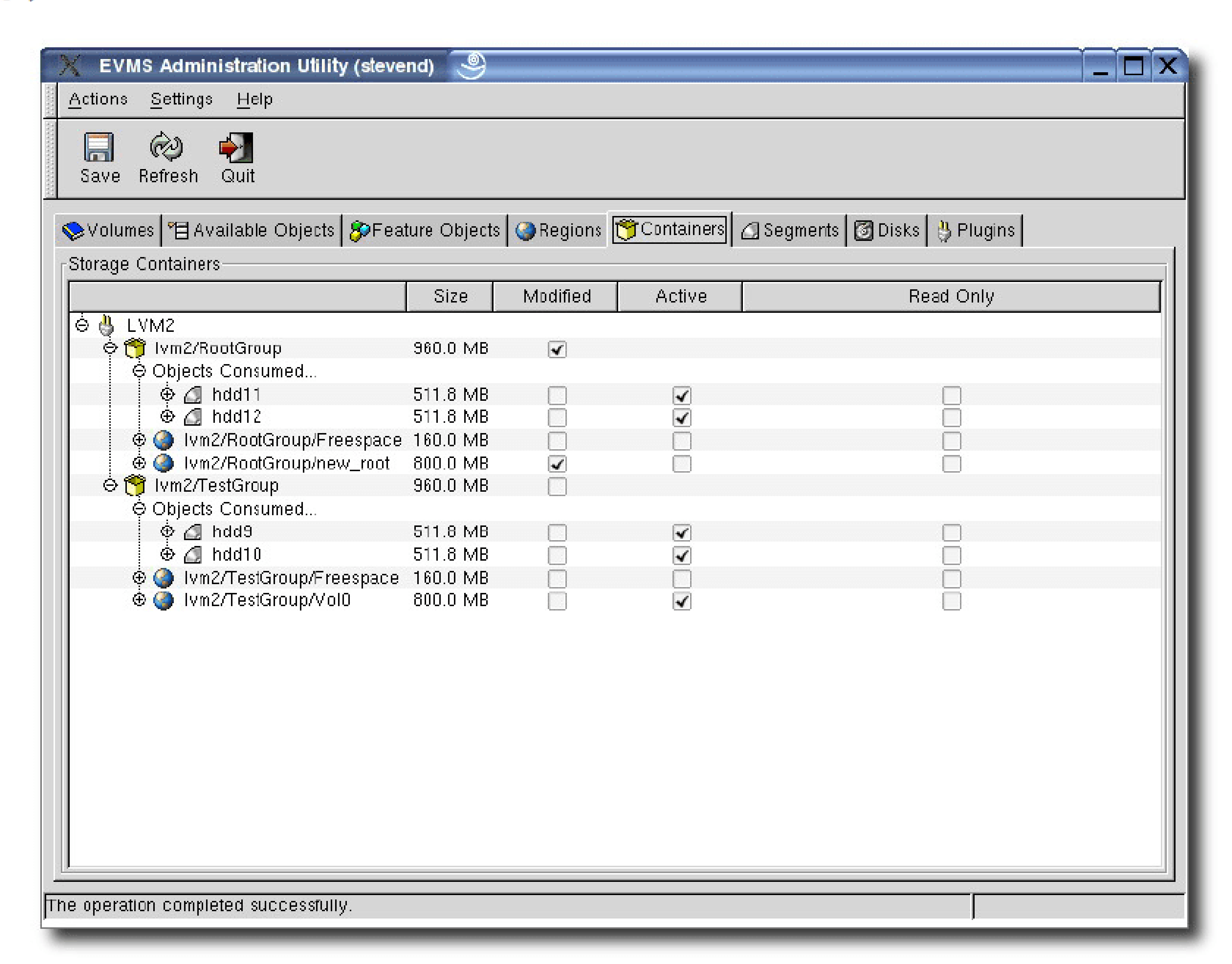


▲ 在 VMware 或是 Hyper-V 中，iSCSI 的使用是最常见的

5.2.2 独立存储设备的底层

一般来说，NAS 和 SAN 的底层一定是使用 RAID 的硬盘数组。组成数组的硬盘，大部分是

相同的通信协议或界面（如 SATA、SCSI 或 SAS）。较高级的硬盘柜可以放入上千个硬盘，总容量常常在数百或数千 TB 以上。这么大的容量当然是被多台计算机使用。以 NAS 来说，只要配置正确的文件系统，就可以使用文件夹分享来访问，但是 SAN 既然能被计算机视为本地硬盘安装操作系统，就必须有“分区”的概念，那么在 SAN 上的分区是怎么做出来的呢？是直接访问单个的物理硬盘，还是直接在动辄上千 TB 的容量上管理呢？这就是 Volume Group 观念引进的最好时机了。



▲ 在创建好的 RAID 之上，存储设备都会使用 VG 的概念

1. 传统 DAS 面临的问题

以 Linux 的硬盘分割为例，我们在安装 Linux 时，必须针对磁盘进行分割。然而在安装之后，经过长期的使用，难保硬盘的容量永远够用。当硬盘满的时候，除了将数据副本导出之外，就是加入新的物理硬盘。当新的硬盘安装上去后经过分割与格式化等过程，最后将新的硬盘分区挂入到另外一个目录内使用，用户必须将这个硬盘挂到另一个新的目录中，如/home2。新的硬盘被挂载到另一个目录来提供后续文件目录放置，虽然有新空间可以用，但必须要各自把文件分别放在和根目录不同的文件夹中，在使用上十分不便。

```
Disk /dev/sda: 8589 MB, 8589934592 bytes
255 heads, 63 sectors/track, 1044 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Disk identifier: 0x0009ad1f

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1   *           1          25      200781   83  Linux
/dev/sda2              26        1044     8185117+  8e  Linux LVM

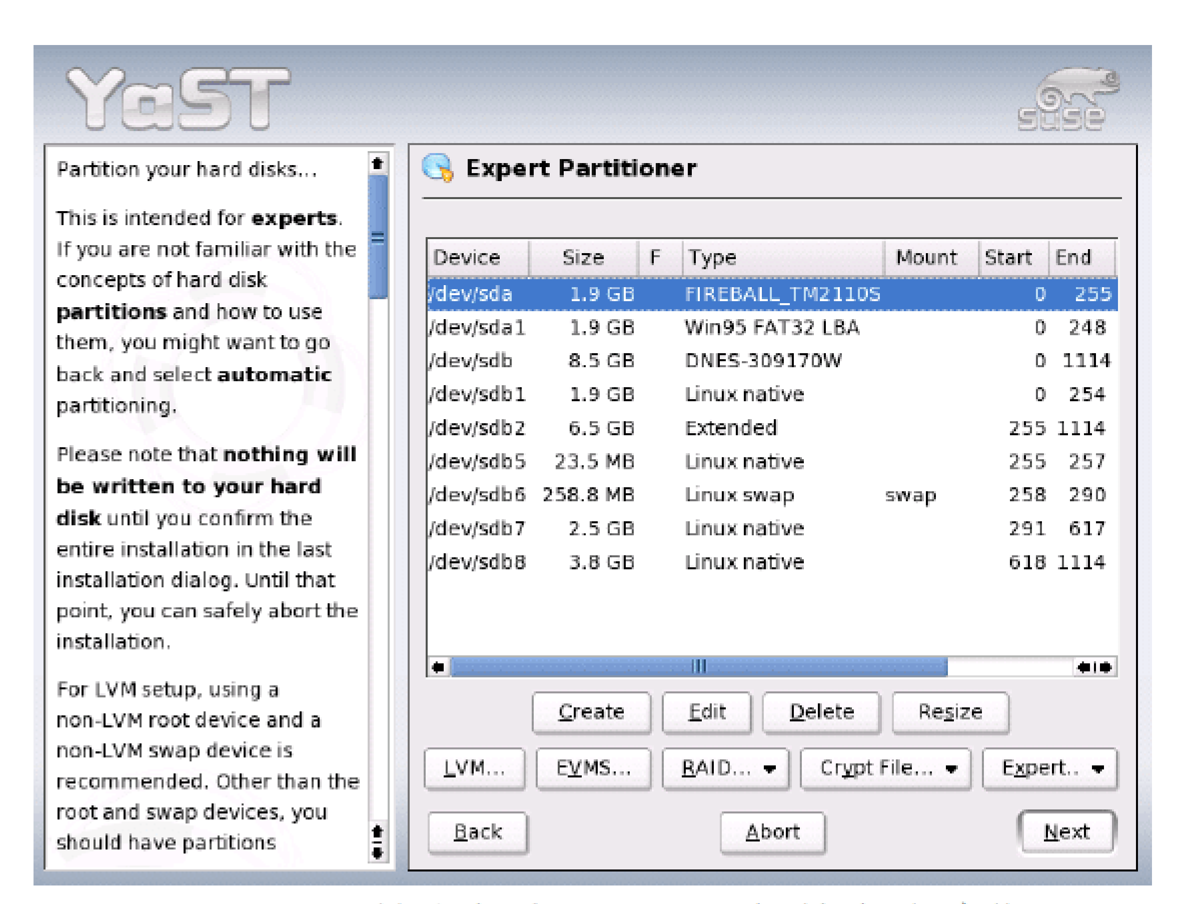
Command (m for help):
```

▲ VG 的专案在 Linux 的安装时就可以直接使用

2. VG 的定义

VG 是逻辑卷宗组的简称，主要的目的就是用来管理多个物理磁盘的观念。VG 可以在硬件

或软件上实现，如 Linux 下的 LVM（Logical Volume Manager）管理程序，主要是 Linux 环境下对硬盘分区进行管理的一种机制。在 SAN 中，也有类似 LVM 的管理程序，其观念就是创建在硬盘和分区之上的一个逻辑层，以提高硬盘分区管理的弹性。

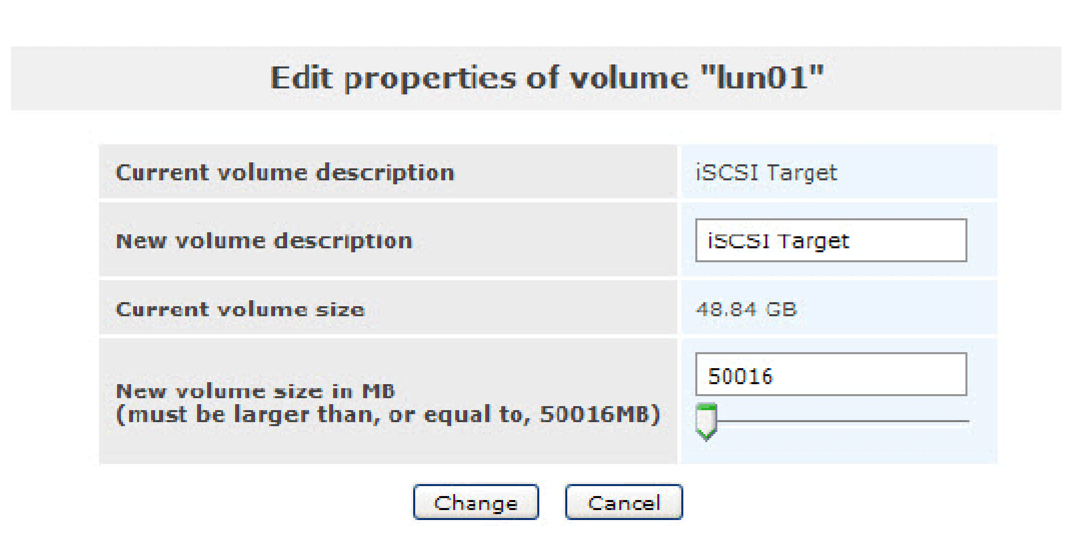


▲ LVM 是大部分 Linux 下都具有的功能

3. VG 能解决的问题

VG 最大的特色就是由不同的物理存储设备组成一个庞大的大空间，系统将这个大空间视为一个大的虚拟硬盘。用户在使用这个虚拟硬盘时，仍然要进行分区的动作。当这个大的虚拟硬盘的空间用完之后，你可以随时加入新的物理存储设备，此时这个大的虚拟硬盘会立即添加空间。以 SAN 来说，如果我们将 500 个 1T 的硬盘组成 500T 的磁盘数组，首先我们会使用 RAID 来进行底层的配置，将整个数组变成一个超大的硬盘池，之后就必须使用 LVM 来进行分区的创建了。

以上面 Linux 的例子来说，若引入 LVM 机制的话就可以很简单的解决问题。若系统使用 LVM 来规划分配系统可用的存储空间，系统的空间分配不再由单纯地单个磁盘分区所提供，而可以是多个硬盘空间合并的结果。而且 /home 可以在持续的加装新的硬盘下就可以扩展空间，可以在不破坏原本数据下就可以进行该功能。因此如果使用了 LVM，在/home 满了的时候，再加入新的物理存储设备到计算机中，此时硬盘就会有多出来的空间，再将这个空间给定到/home 之下，立马就可以让/home 直接使用新的硬盘了。



▲ 使用 VG 的好处是能无限扩展空间，只要再加入新的硬盘就行了

5.2.3 VG 的架构图

VG 也是一个分层的架构，并且对于 SAN 来说，VG 是完全透明的。换句话说，在进行硬盘池的管理时，系统并不会认为这是一个特殊的磁盘，而会将其视为一个物理磁盘，在安全性、效

能和弹性上有相当大的进步。接下来我们就来看看 LVM 的架构。

▲ 这是标准 VG 的架构图，图中的硬件层并没做 RAID，标准的是有 RAID

1. 物理存储设备（Physical Media）

VG 的底层仍然是硬盘，但是这个硬盘已经是经过 RAID 或是 NAS/SAN 制造出来的超大硬盘了。VG 底层的“物理硬盘”甚至可以是 SATA、SCSI、SAS 不同存储设备的任意组合，只要是能被 RAID 识别出来的存储设备，都可以作为 VG 的物理存储媒介。

2. 物理分区（Partition）

存储媒介是可以再分区的。以 Linux 下的 SCSI 硬盘来说，一个硬盘被称为/dev/sda，第二个 SCSI 被称为/dev/sdb。如果你将第一个硬盘分割成两块，就会有/dev/sda1 和/dev/sda2。这些都是物理上的分区。不过在 VG 下，我们一般不会再将存储设备分区了，因为对 VG 来说，最后都会变成一个大的硬盘，底下分得再细都是多此一举。

3. 物理卷（Physical Volume）

当存储设备被给定给 VG 用的时候，所有的分区都变成了物理卷。物理卷和物理分区最大的差别，就是在 SAN/NAS 下，只有被给定成物理卷的分区才能给 VG 使用，没有被给定成物理卷的分区，仍然被视为一个独立的硬盘分区。

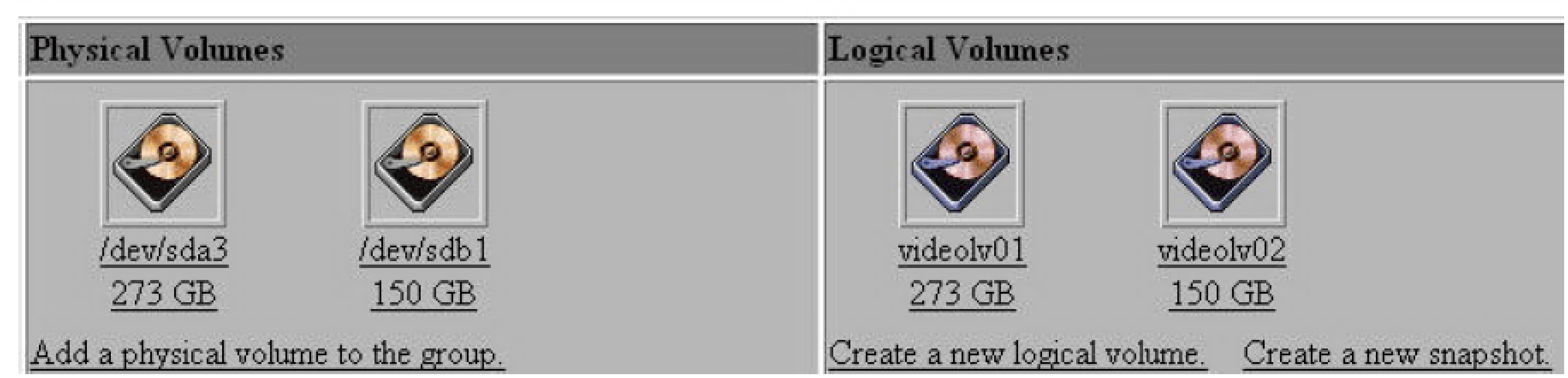
4. 卷组（Volume Group）

VG 就是整个 SAN 架构中最底层的硬盘空间了。一个 VG 是由许多物理卷所组成的，对 Linux 系统来说，这是一个硬盘空间池，包括针对硬盘的分区、文件系统、目录结构等，都是操作在 VG 之上的。因此在配置了 VG 之后，用户还是需要在 VG 这个“新”的硬盘空间上做分割，安装文件系统，才能继续安装其他的文件系统。

5. 逻辑单位（Logical Unit Number，LUN）

LUN 是 SAN 中最重要的一个单位，也是所有计算机访问 SAN 的基本单位。在软件的 LVM 中，也称为逻辑卷（Logical Volume，简称 LV）。LUN 就是我们在 VG 上面做的分区。讲到这里我想读者们都有一个思路，就是这么做是不是有点多此一举。我可以直接使用原来硬盘的分区，为何要将分区组成再拆开呢？使用 LUN 的好处包括了弹性、管理以及不停机等，我们会在本章

稍后说明。



▲ LUN 是所有 SAN 访问中最重要的单元

6. 文件系统（File System）

创建了 LUN 之后，就可以在 LUN 上创建不同的文件系统（简称 FS）了。这里所说的文件系统就是计算机使用的文件系统，如 Linux 下的 ext3 或 Windows Server 2008 下的 NTFS。对操作系统来说，LUN 就是一个物理硬盘，既然是透明的，因此文件系统也不会管这是真正的物理硬盘还是 LUN，计算机在引导时，会先通过 HBA 界面卡获取 LUN 的信息，并且被视为本地硬盘，就如同 SATA 界面卡抓到 SATA 硬盘或是 SCSI 卡抓到 SCSI 设备的意义完全一样，之后在装载驱动程序之后，就可以被操作系统认出开始安装了。

7. 物理存储单位（Physical Extent）

在 VG 创立的过程中，每一个分割也有最小的存储单位，称为物理存储单位（简称 PE）。PE 是每一个物理卷的存储小格，LUN 中的每一个数据都是由 PE 所组成。这个值在创建文件系统时是可以更改的。

8. 逻辑存储单位（Logical Extent）

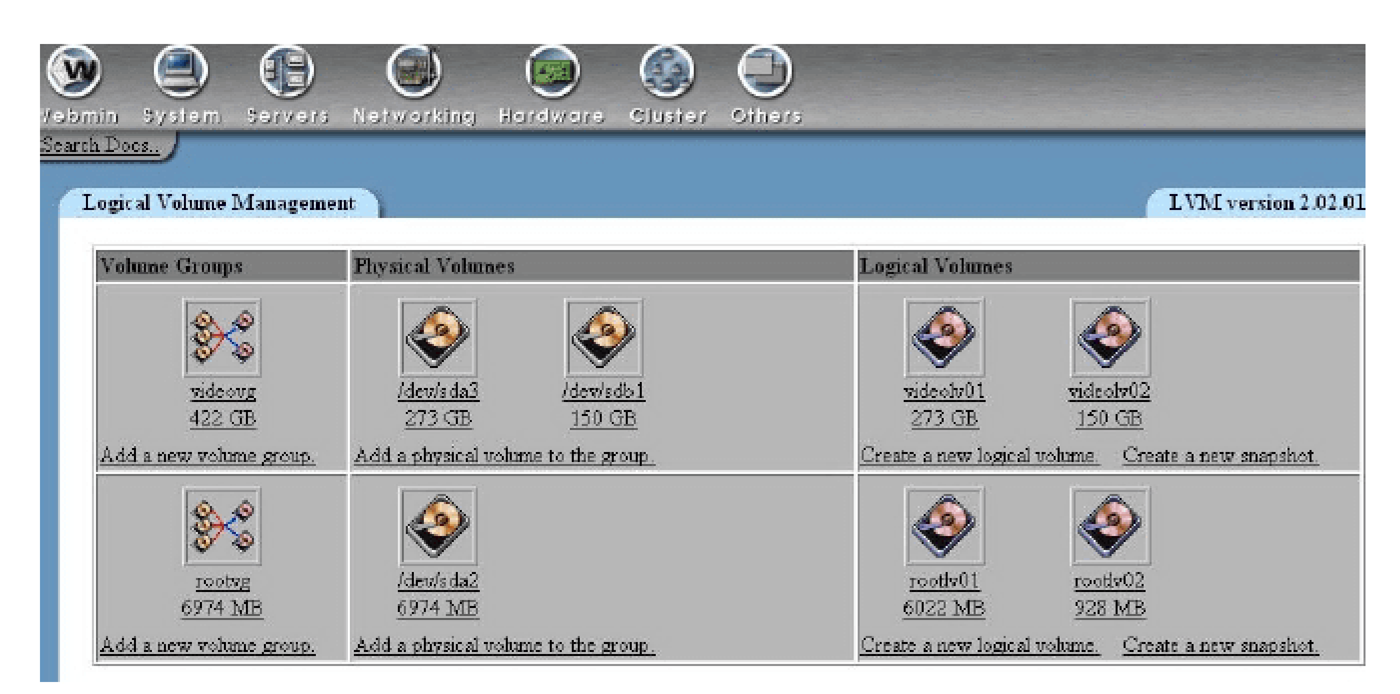
对应到 LUN 上，在逻辑分区上的最小存储单位就被称为逻辑存储单位（简称 LE）。LE 是 VG 系统中可被寻址的基本单位。在同一个卷组中，LE 的大小和 PE 是相同的。

5.2.4 使用 VG 的优点

VG 在管理、安全、弹性以及系统的高可用性上是 FC SAN 或 NAS 能成为虚拟机产品青睐的主要原因，大部分的 SAN 产品都使用了 VG 的方式来管理硬盘，而大部分以 Linux 为主的 NAS 产品也使用了 VG 及 LVM 观念来管理硬盘数组，我们就来看看其优点。

1. 分区能有更大的弹性

VG 的第一个优点，就是可以拥有弹性大小的分区。举例来说。系统安装了 500G 的 SATA 硬盘，300G 的 SCSI 硬盘，以及 200G 的 IDE 硬盘。在这个系统中，能创建的最大分区就是 500G（即最大物理硬盘的单个分区）。但是如果用 VG 的观念，将这三个存储设备合并成一个 1TB 的 VG，能创建的分区最大就是 1TB 了。以一个使用 Linux 的 NAS 来说，留 2GB 给置换档用，留 200MB 给/boot 用（不建议将/boot 放在 VG 中，放入物理硬盘中即可），剩下全部给根目录，根目录可以使用将近 998GB 的空间。如果没有使用 LVM，最大就是 500GB，作为一个影音或是文件服务器来说，的确没什么弹性。



▲ 在 VG 中，我们可以任意分割磁盘而不受到物理硬盘大小的制约

2. 管理上的方便

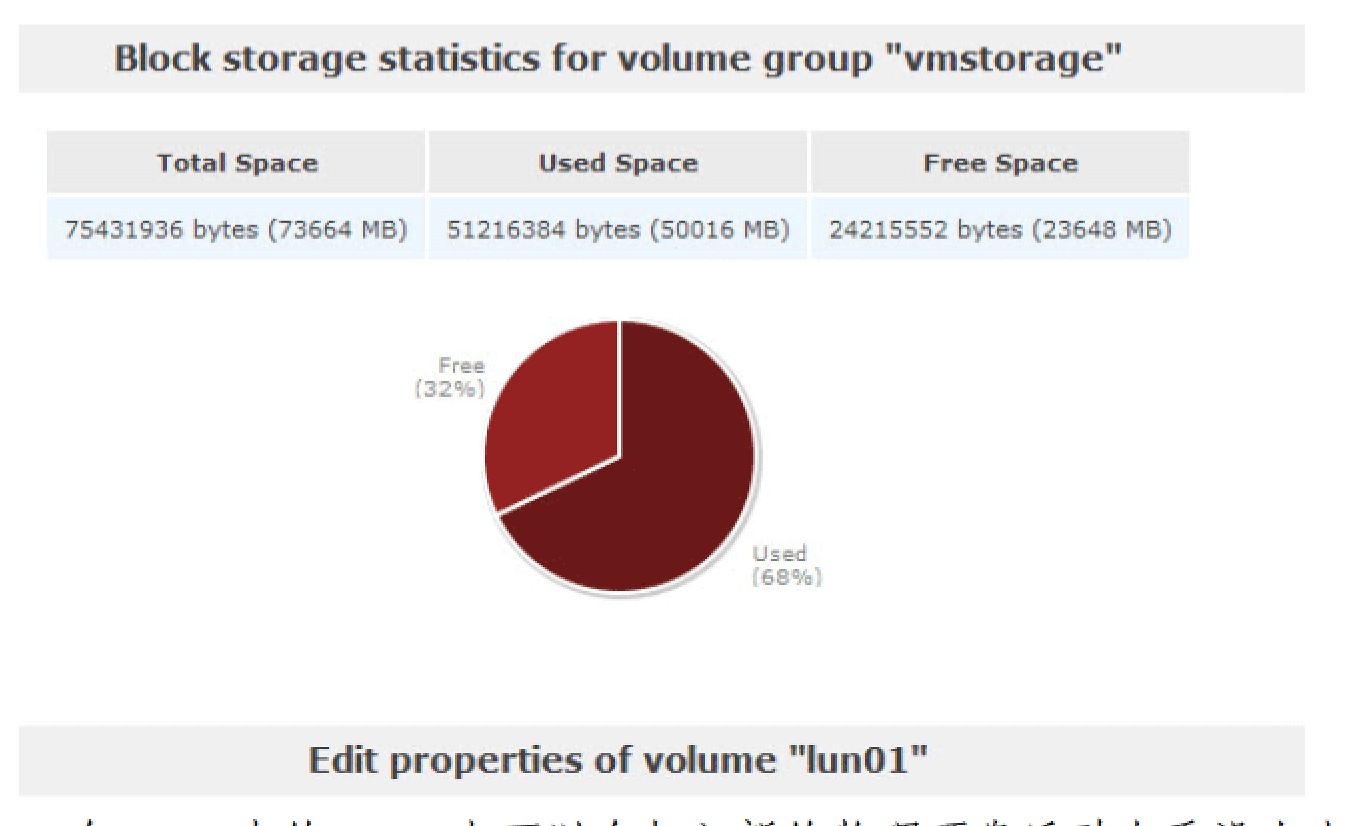
使用 VG 另一个好处就是管理上的方便。在 Linux 下，所有的硬盘都被赋予 Linux 内定的名称，如前面提到的/dev/sda、/dev/hdb 等。对管理员来说，必须制作一个对应表才能知道每一个分区是对应到哪一个目录。但用了 LVM 之后，你可以将不同的分区命名为“/sales”、“/development”、“/market”之类的合理名称，当其他的人需要管理 Linux 主机时，能一眼就知道这个分区的意义。



▲ 使用有意义的名字而非 Linux 下的设备名称能添加管理时的方便

3. 可动态调整分区大小

在 VG 中使用逻辑卷组，并且进一步在逻辑卷组上创建文件系统之后，管理员通过 LVM 可以方便地调整存储卷组的大小，并且可以对硬盘存储按照组的方式进行命名、管理和分配。如果空间不够，直接将物理硬盘加入这个 VG 中，到新的硬盘上分配，直接扩展文件系统跨越硬盘即可。



▲ 在 SAN 中的 LUN 也可以在加入新的物理硬盘后动态重设大小

4. 不需要关机操作

现在新的 SATA 硬盘或是 SCSI 硬盘都支持热切换（Hot Swap）。在新加硬盘时不需要重启，再配合 LVM 的功能之后，当你发现硬盘的空间不够了，可以直接将新的硬盘插上计算机，使用

LVM 的管理程序来将硬盘加入 VG 中，并且规划空间，整个过程一直不用关机，并且所有的功能都常规操作，对商用网站来说，系统的可用性比什么都重要，LVM 的弹出可大幅提升 SAN/NAS 设备的不中断操作时间，减少可预期停机起到的损失。

5.2.5 VG 在 SAN/NAS 下的使用时机

所有的 NAS 或 SAN 设备厂家都有自身的 RAID/VG 机制，大部分都是通过专属的程序进入管理，而多台磁盘数组也可组成更大的 VG 组。针对虚拟机环境来说，独立存储设备的使用通常包括下列几个场景。

1. 需要管理多台存储设备

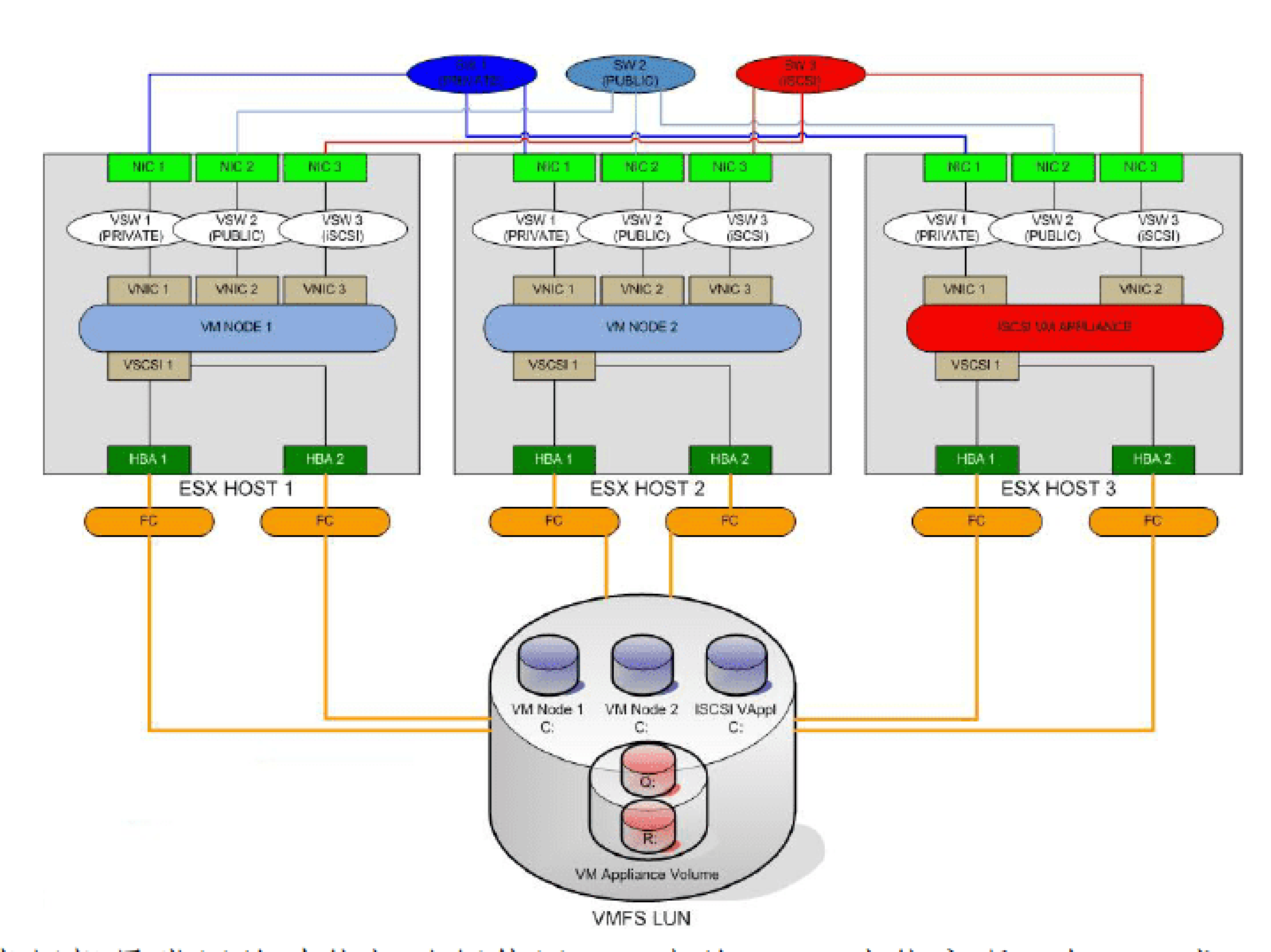
在较正规的商业环境中，系统管理员使用的主机很可能是有不同的存储介质，包括了服务器最爱用的 SCSI 硬盘，或是现在售价相当便宜的 SATA 硬盘，或是由以上两种不同硬盘组成的 RAID，当然还有网络存储系统 SAN 等。使用 VG 上的 LVM 可以让系统管理员很方便将不同的存储媒介组成一个大的 VG 或是几个不同的 VG，并且将不同的 LUN 作最适当的应用。

2. 硬盘空间快捷使用

这种情况多半发生在影音的数据库、副本用服务器，或是文件、FTP 服务器上。当前影音数据相当流行，大型网站上都存储了许多电影或是 MP3，再加上网络带宽的发达，计算机一天下载几 G 的数据并不罕见，此时在硬盘空间快捷消失的情况下，必须常常新建硬盘，这时候使用 VG，可以在不改变整个 SAN/NAS 的磁盘结构、不用重启的情况下落实这个任务。

3. vSphere 的 HA 环境

使用虚拟机的企业必须将独立存储设备规划好，不同的系统管理员可以随时针对企业的需求对存储做动态调整，并且在不太需要制作说明文件的情况下，让新的系统管理员快捷接手。这时如果使用 VG 上的 LVM，可以用更人性化、更直接的方式来管理 SAN 上的 LUN，让整个管理流程更为顺畅。



▲ 虚拟机最常用的功能都必须使用 VG 上的 LUN 才能实现，如 HA 或 VMotion

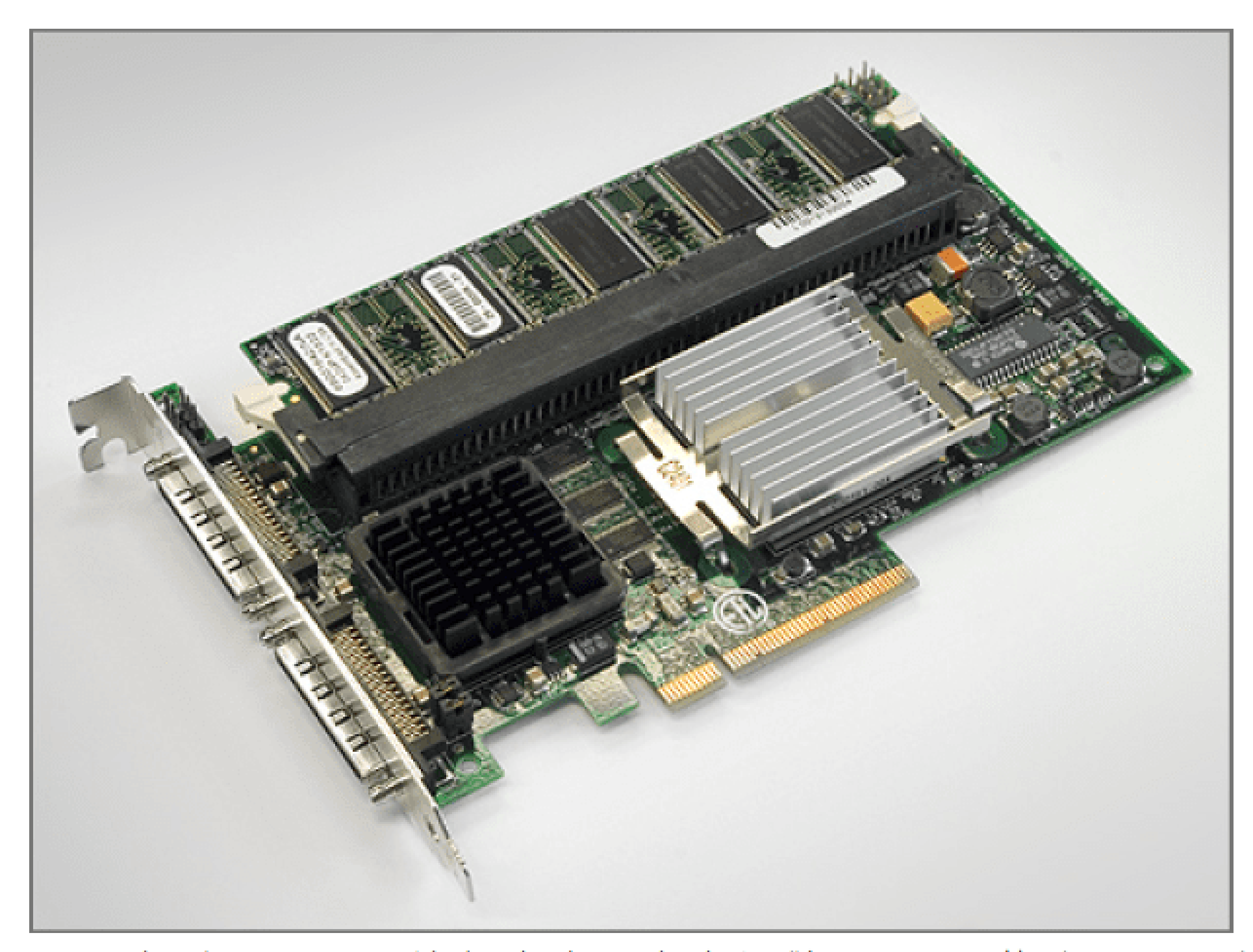
5.2.6 VG 与 RAID 的不同

近年来，由于硬盘价格大幅下滑，再加上 SATA 界面的流行，RAID 这个从前只能在 SCSI 设备上看到的设备，也悄悄走入个人计算机的使用了。RAID 的原理也是将许多硬盘组成一个大的硬盘，并且在其上做分区，听起来和 VG 完全一样，但在原理和使用上是完全不同的。

1. RAID 主要以安全和性能为主

RAID 主要的目的在于实时副本，并且提供 XOR 的遍历机制，因此在 RAID 硬盘上，除非你做成 RAID 0，要不然所有的数据会有遍历数据，分布在所有的物理硬盘上，当任何一台硬盘出毛病时，都可以利用 XOR 遍历数据将坏掉的数据还原，并且在经过重建（rebuild）之后，将系统恢复成原来的样子。

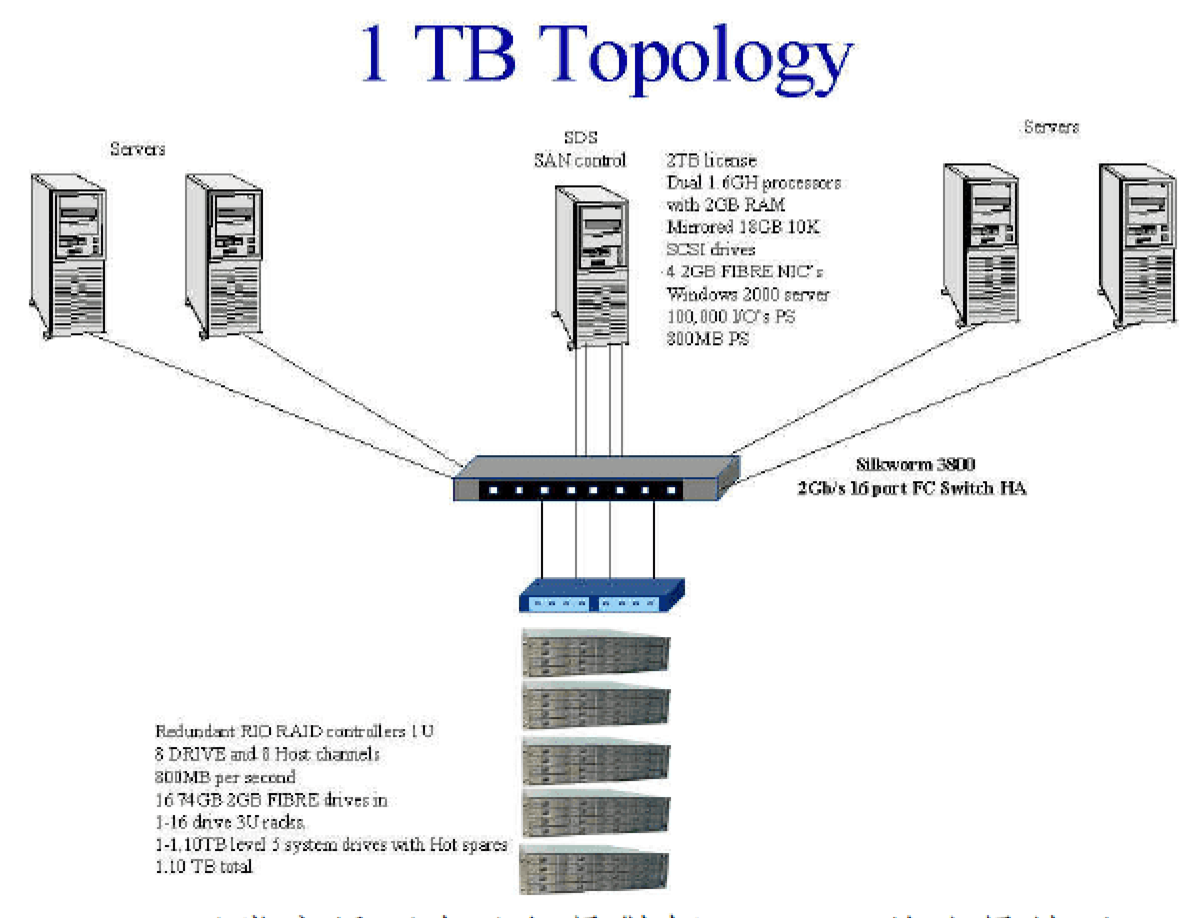
就算是 RAID 0 做成了连续磁盘，主要也是为了性能问题，任何数据都被拆成两份，存储在不同的硬盘上，不但在可写时可以加快一倍的速度，在可读时也可以加快速度。



▲ 正规的 RAID 环境都会有一个专门做 XOR 运算的 RAID 卡

2. RAID 是运作在硬件层

除了软件类比的 RAID 之外，所有的 RAID 都是运作在硬件的层级，换句话说，你必须有 RAID 卡，或是主板上有 RAID 的芯片组，才能将很多硬盘做成 RAID。当 RAID 硬盘落实之后，对 Linux 来说，在硬件层级就是一个大硬盘了，甚至连操作系统都不知道这是 RAID。但对 VG 来说，是在 NAS 管理系统中制作的，运作在软件层，因此 VG 的物理硬盘通常创建在 RAID 存储设备之上。



▲ 通常底层硬盘已经是做好 RAID 5 的大局块了

5.3 具体操作存储设备

在理解 RAID 及 Volume 之后，接下来我们就要在物理设备上使用这些东西了。物理设备分成 DAS、NAS 及 FC SAN 或 iSCSI 等。DAS 的处理一般是在本地安装 ESX/ESXi，我们在安装的章节会有说明，而 FC SAN 及 iSCSI 则会在创建独立存储及 RDM 的章节会有说明，我们接下来，就以一个 NAS 作为例子来说明 RAID 以及 Volume 的创建。

5.3.1 使用 NAS 设备

当前大部分的公司都有独立的存储设备，最被广泛使用的就是 NAS 了。NAS 如第 4 章所提，是一种使用网络连接的存储设备。较高级的 NAS 除了能作为网络磁盘机分享之外，更有 iSCSI 以及 LUN 等高级功能，本章就以一个具有多网卡，能提供 iSCSI Target，并且能插入 8 台 SATA 硬盘的高级 NAS 进行存储设备的配置示例。



▲ 当前大部分的公司都有 NAS 产品，图为机架式的 NAS 产品

1. QNAP TS-859Pro 简介

QNAP 是 NAS 设备的著名生产商，其产品主要是以真正市售 SATA 硬盘组成的磁盘数组为主。QNAP TS-859Pro 是一个 8 bay 的 NAS 产品，和一般市售 NAS 最大的不同，就是 TS-859Pro 具备了 iSCSI Target 的功能，让 vSphere 的 ESX/ESXi 都能直接找到产品而挂载上，让以成本考虑的公司还能使用到 ESX/ESXi 的高级功能（如 RDM）。此外如 NFS 或是 RAID、Volume，在 TS-859Pro 上都能很轻松管理，让你对 vSphere 的使用真正落实全功能的实践。



▲ 塔式的 NAS，本身拥有相当完善的功能

2. 将 NAS 连上 vSphere 环境使用

NAS 是一个网络设备，因此必须连上网络环境使用。TS-859Pro 和大部分的 NAS 一样使用了一个网页界面来管理。虽然 TS-859Pro 也有 Console 界面，但必须连上 COM PORT，或是使用最简单的面板来管理，我们在本书中的演示，都是以网页界面的管理方式操作。



▲ 大部分的 NAS 都是使用浏览器来配置

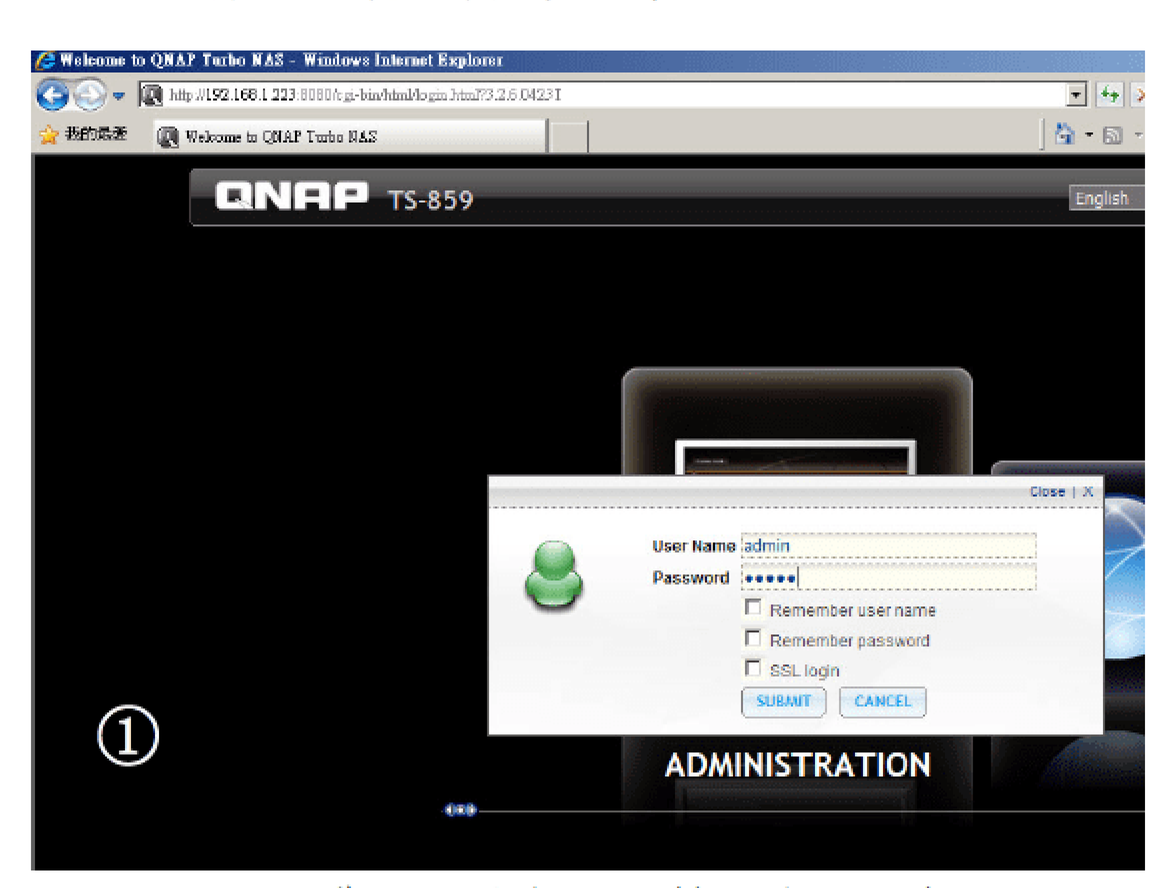
5.3.2 利用 TS-859Pro 创建 RAID 组

在 vSphere 中，使用独立存储设备的目的就是使用 VMotion 等功能。因此着重的一定是安全而非速度，因此几乎没有人会在 vSphere 环境中的独立存储设备使用没有安全性的 RAID 组合（如 JBOD 或 RAID 0）。我们在这一小节，就来演示使用 TS-859Pro 做出来的 RAID 1+HowSpare 以及 RAID5。

▶ RAID 1 的制作

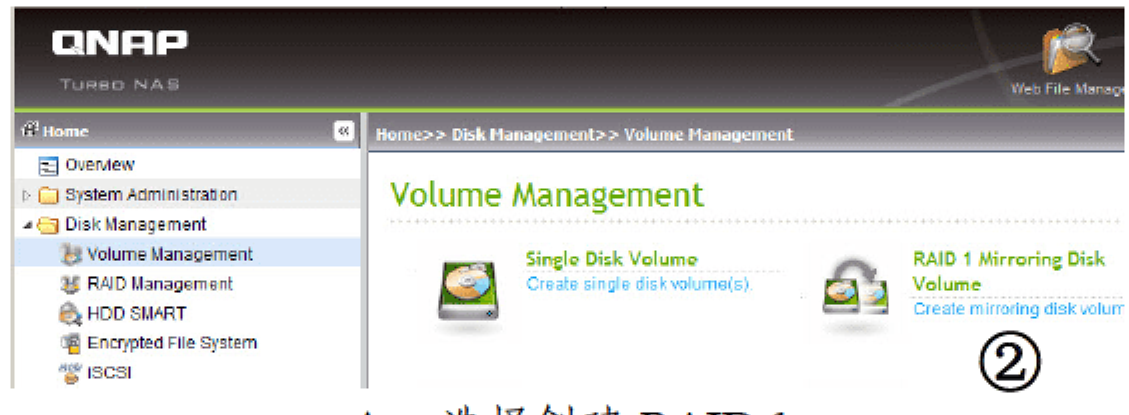
RAID 1 需要 2 个硬盘，我们就以 2 颗硬盘作为实例做成 RAID 1。

1. 从浏览器进入 NAS 的 IP，键入账号口令。



▲ 进入配置的 IP 后键入账号口令

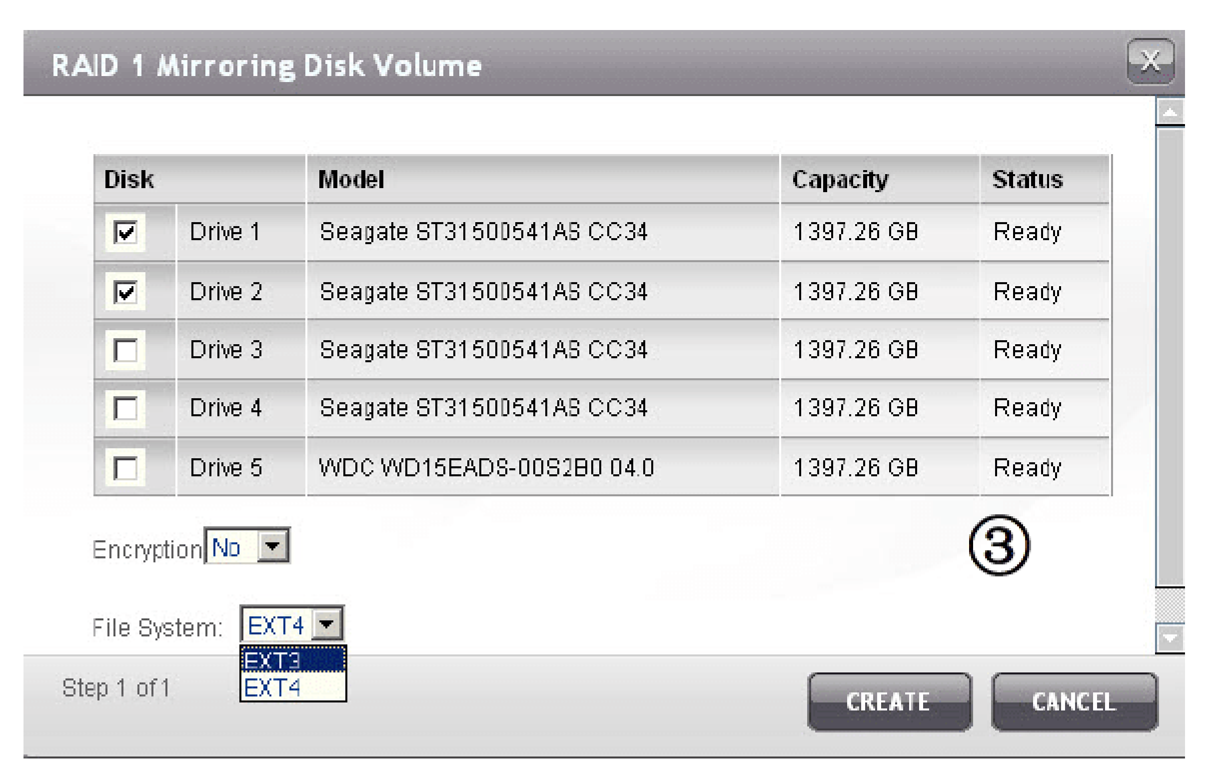
2. 选择 Volume Management，选择 RAID 1 的部分。



▲ 选择创建 RAID 1

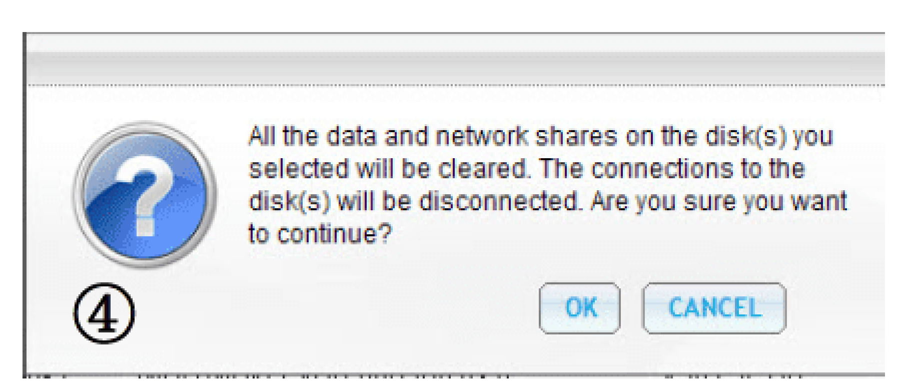
3. 此时我们选择任何 2 个硬盘，文件系统选择 EXT4，这是因为这台 NAS 只支持 EXT3/4

的 Linux 文件格式。单击 CREATE 按钮。



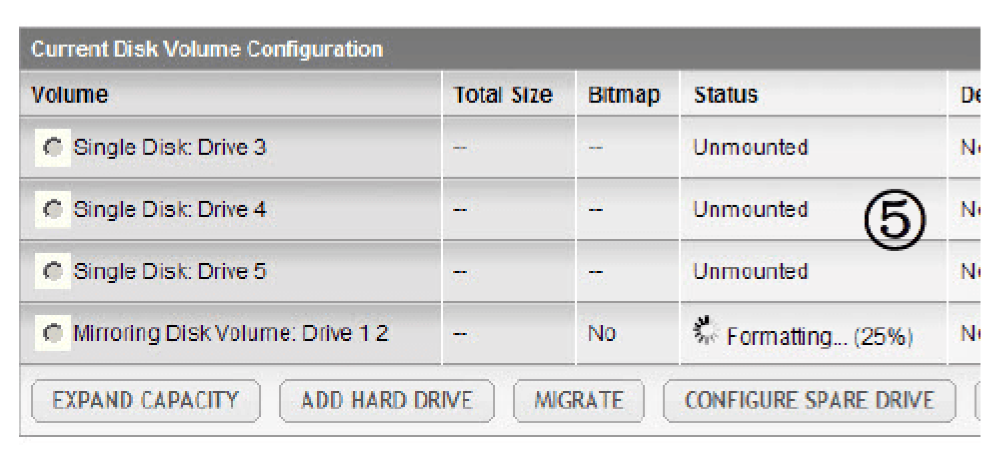
▲ RAID1 需要 2 个硬盘

4. 此时会弹出报警标志，说明 2 颗硬盘上的所有数据都会不见，单击 OK 按钮。



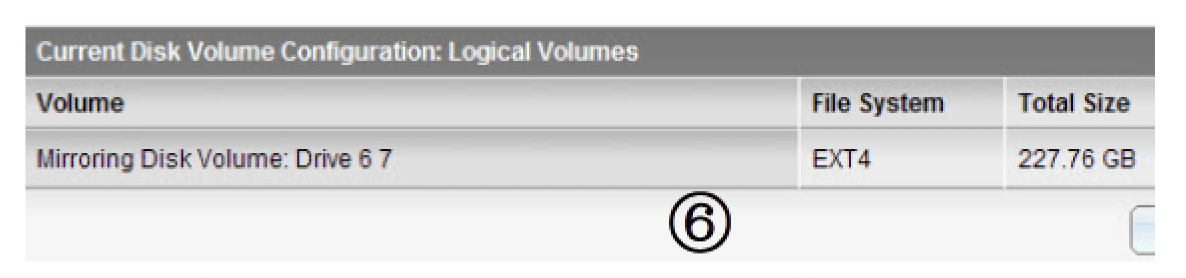
▲ 报警窗口

5. 在网页底部可以看到硬盘已经在进行格式化了。



▲ 在进行格式化了

6. 当落实之后，可以在 RAID Management 中看到有一个 Mirror 的 RAID 组合，就是我们创建的 RAID 1，也是这台 NAS 的第一个 Volume。



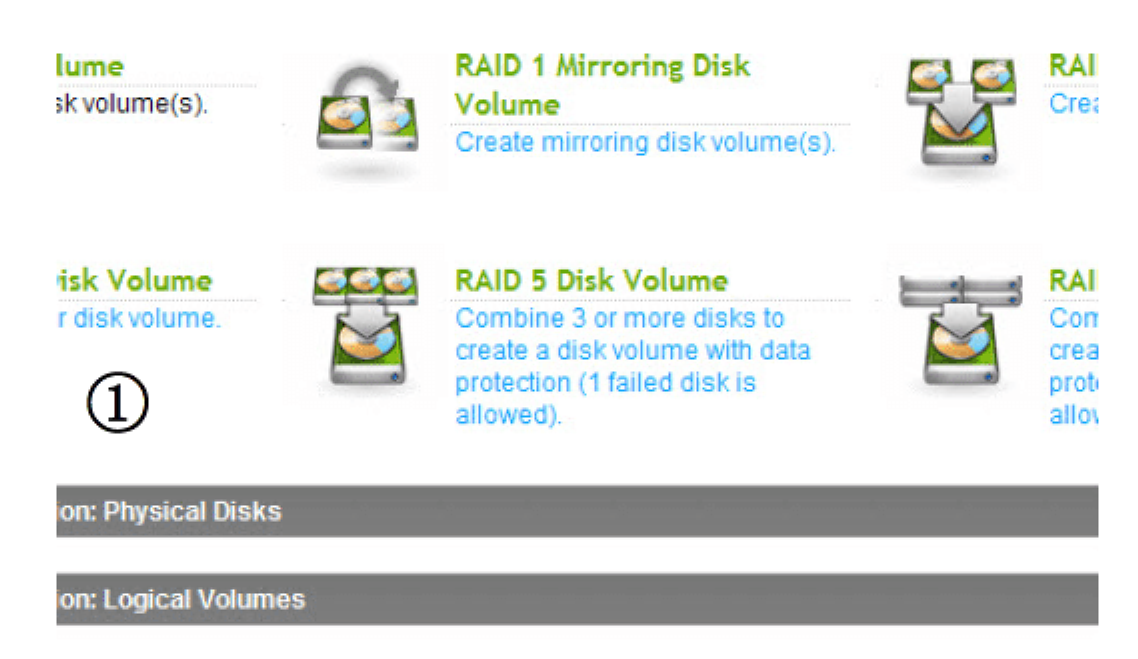
▲ 创建完毕后，是这个 NAS 的第一个 Volume

▶RAID5 + Hot Spare

一般来说，如果有超过 3 颗硬盘时，我们会选择 RAID 5 来达成容量、安全以及性能的平衡。我们就以 5 颗硬盘，来实践 4 颗硬盘的 RAID 5 加上一个 Hot Spare 的实践。

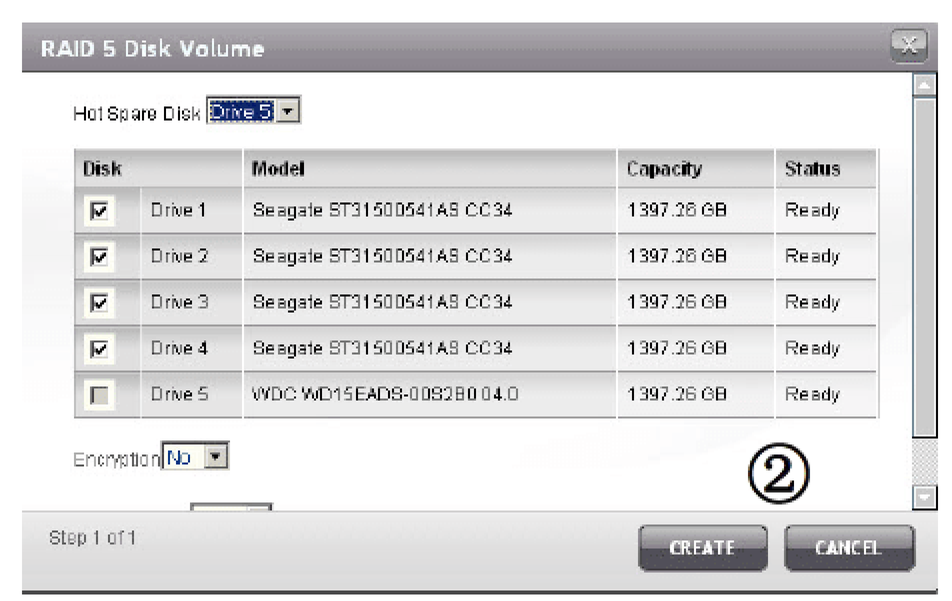
▶ RAID 5 + Hot Spare 的实践

1. 还是进入 QNAP 的菜单，选择 Volume Management。从中选择 RAID 5 的部分。



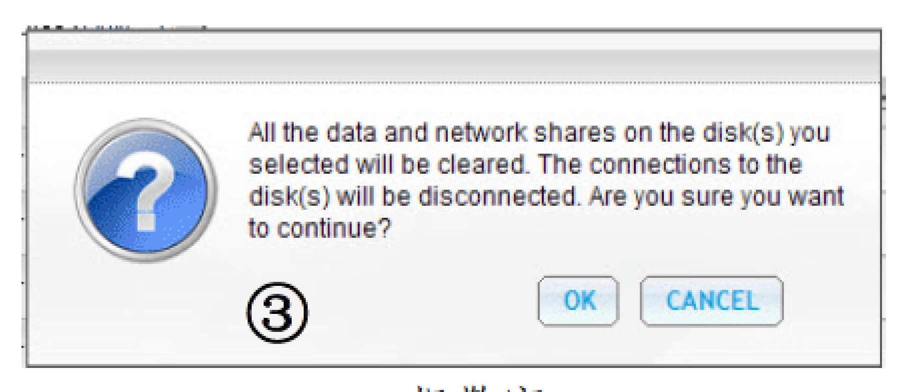
▲ 选择 RAID 5 的部分

2. 选择要加入 RAID 5 的硬盘，我们选择 1～4 颗硬盘，将第 5 颗作为 Hot Spare。然后单击 CREATE 按钮。



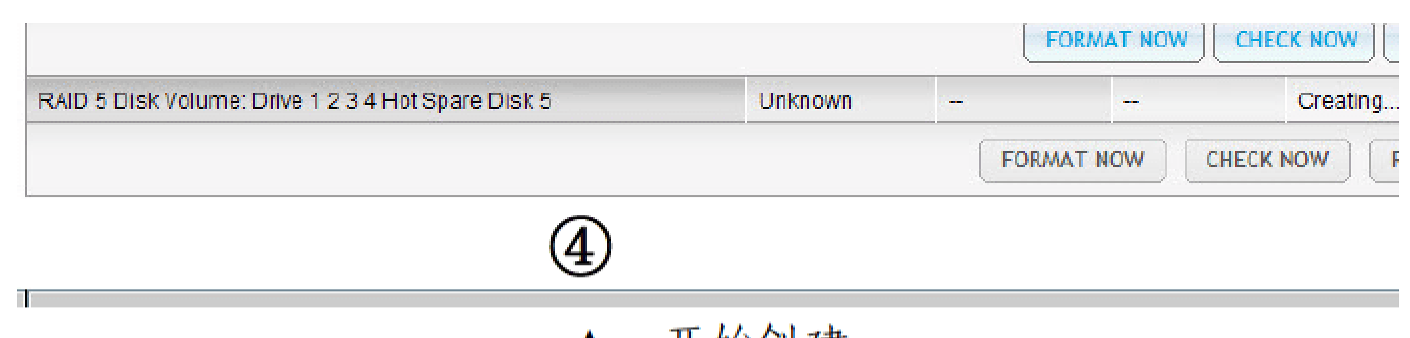
▲ 5 个硬盘做 RAID5，其中一个做 Hot Spare

3. 系统也会弹出报警窗口，单击 OK 按钮。



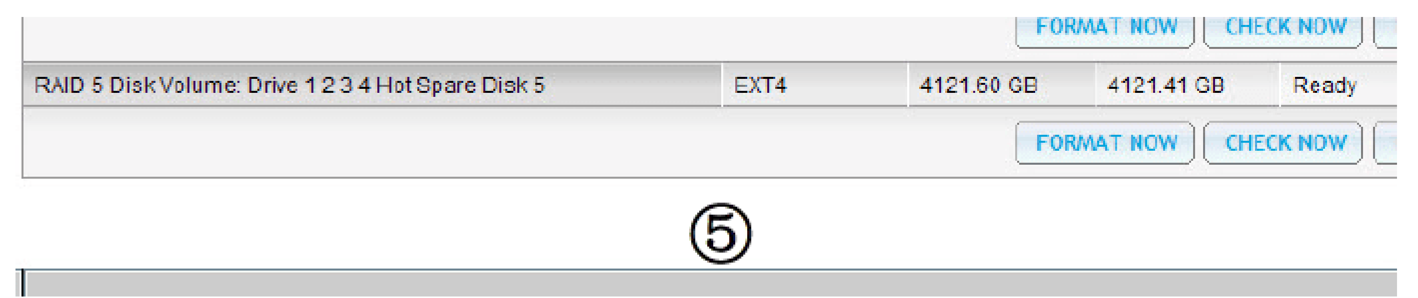
▲ 报警窗口

4. 接着开始创建。



▲ 开始创建

5. 创建完毕之后可以看到这个 NAS 中已经有第二个 Volume 了。



▲ 第二个 Volume 也落实，5 个硬盘做 RAID 5，其中一个做 Hot Spare，真正的容量只有 3 × 1.5TB = 4.5TB

结 语

在 vSphere 下，操作存储设备是最重要的一个任务，而独立存储设备更是虚拟化的基石。在理解了 RAID、Volume 的概念之后，vSphere 下的存储设备应该就没有太大的问题。我们在本书稍后的章节中，会完整介绍 ESX/ESXi 下 iSCSI 的实践，并且会使用到 ESX/ESXi 中最重要的功能 RDM。

第 6 章 虚拟化的性能评估

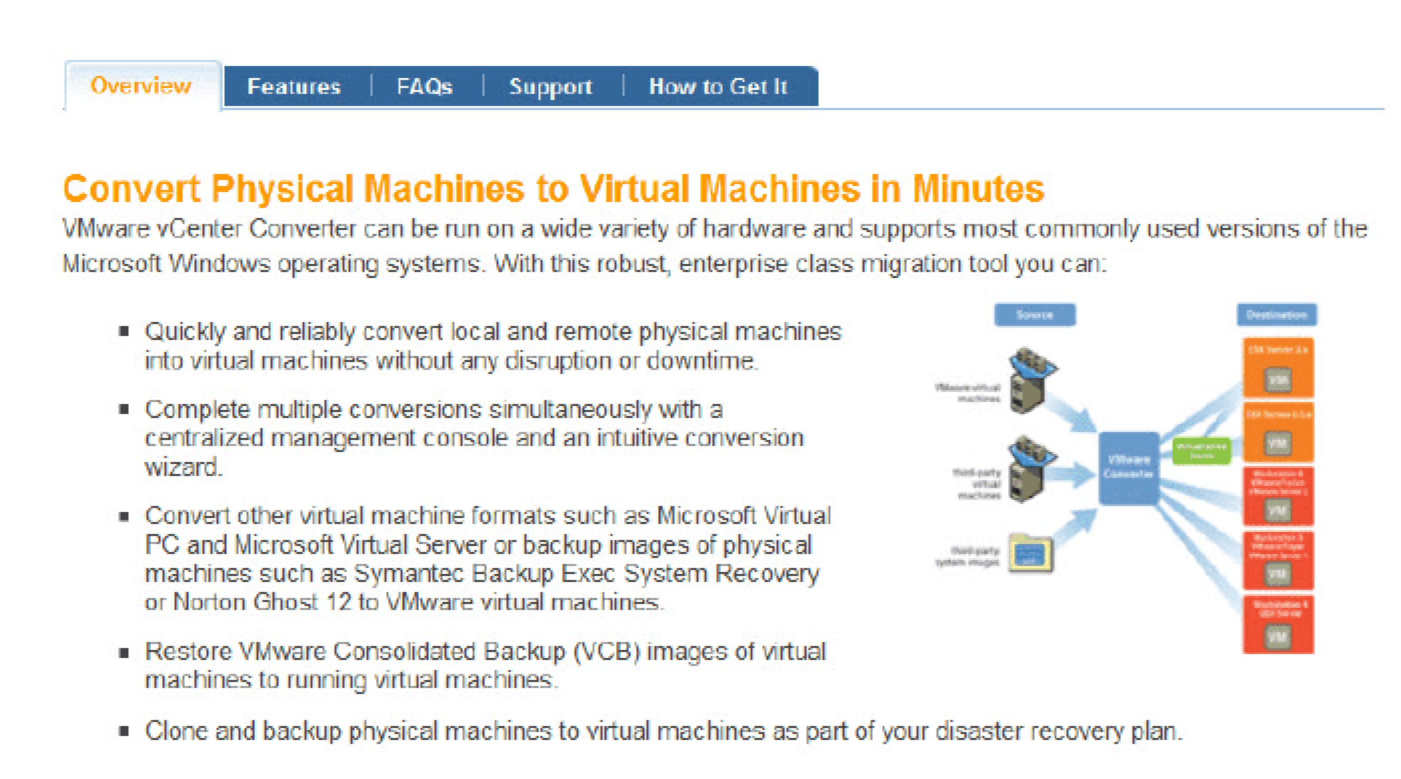
关键词：

- Windows 下的效能监视工具
- 使用 PAL 来监视 Windows 下的服务效能
- 使用 MRTG 来监视 Linux 下的网络环境
- 使用 MRTG 来监视 Linux 下的 CPU 使用

在 vSphere 中，最让人担心的就是企业使用虚拟化环境之后，会不会有效能上的问题。由于每一种应用的情况不同，因此难保上了虚拟化之后会发生资源短缺的问题，因此在真正将企业的 IT 环境转换之前，理解每一个服务器上服务的效能“瓶颈”是最重要的。一般来说，CPU、内存、I/O 性能（存储卡及网络）是我们最在乎的几个重点。在进行转换前，使用专业的工具来提取效能是最正确的做法，我们在这一章，就来看看 Windows 下和 Linux 的效能提取。

6.1 Windows 系统效能的评估

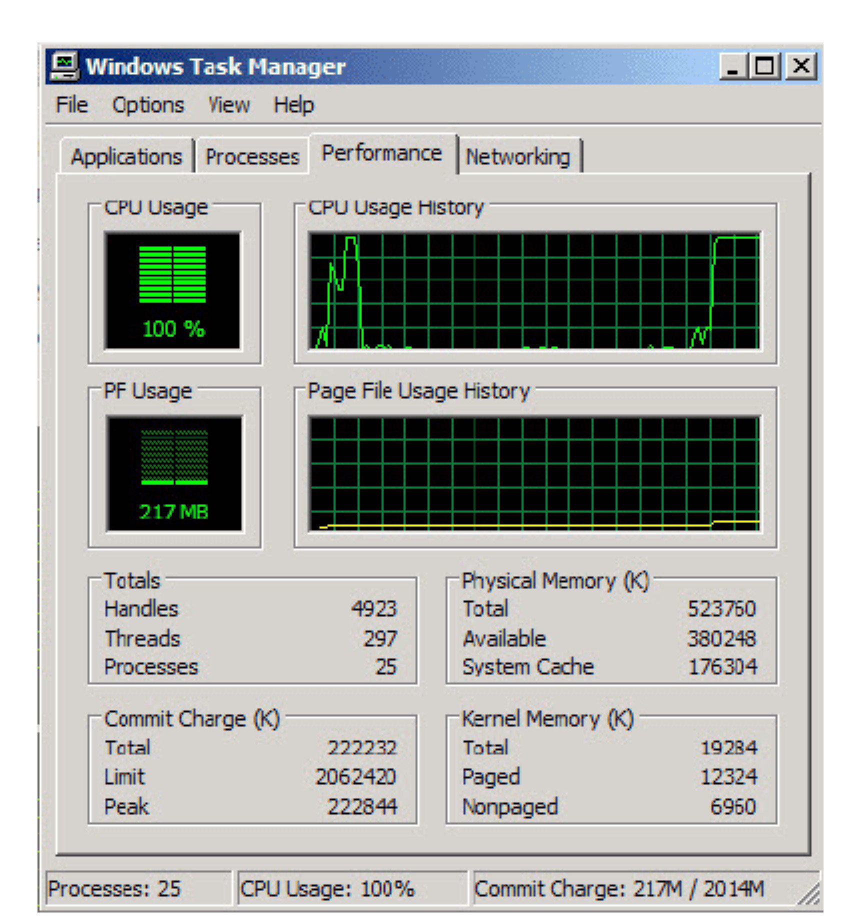
在理解一台常规使用的服务器效能之前，贸然进行虚拟化是相当危险的。由于虚拟化的过程是将物理机转换成虚拟机，因此物理机的硬件效能不够强大，或是虚拟机的服务占用太多系统资源，都会影响 IT 环境中的其他物理/虚拟机，因此我们在本章先来看看评估效能的量化方式。



▲ 较著名的虚拟机厂都提供 P2V 工具将物理机转上虚拟机

6.1.1 正确评估硬件 CPU 使用效率

虚拟化的前提是硬件的利用率不佳，当我们知道不同的服务器的平均使用率之后，我们就可以决定一台物理机上能使用多少个虚拟机了。决定一台物理机能使用多少个虚拟机的因素很多，但在将存储的负载移到 SAN 上面之后，最重要的就是 CPU、内存和网卡。由于内存可以用直接对应方式来安装，而网卡则是以服务的种类而定，因此最重要的就是 CPU 了，我们就来看 CPU 的效能查看方法。



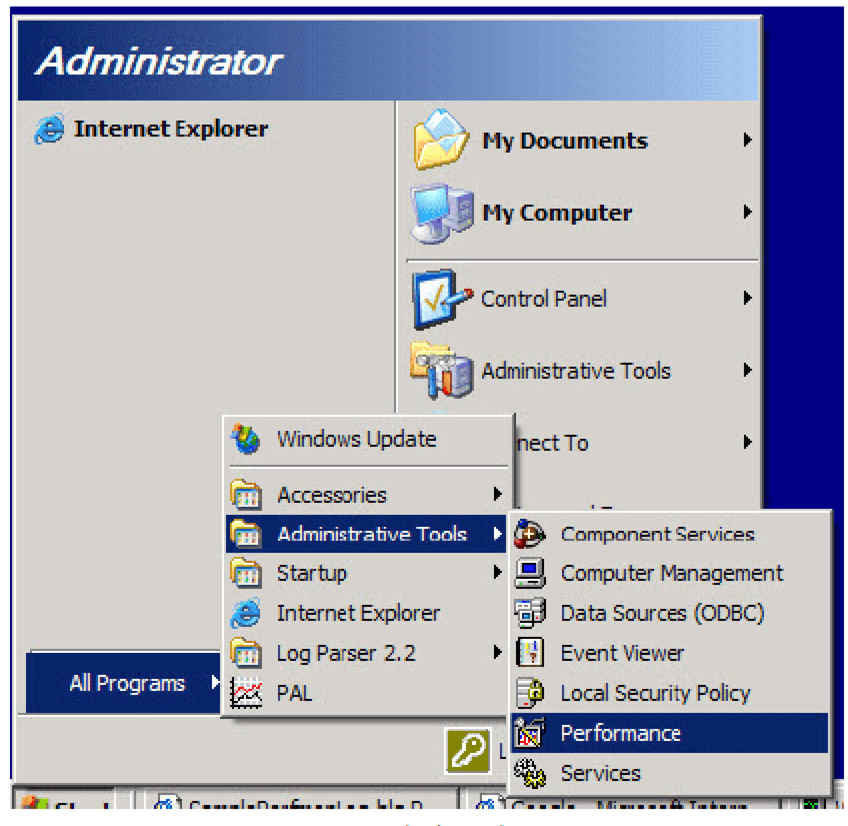
▲ 一个满载的服务器是不适合移到虚拟机上的，尤其是 CPU

1. Windows Server 2003/XP

在 XP 和 Windows Server 2003 中有一个“效能”的功能。这个功能可以将 CPU 的使用率持续表项下来，为了理解整个服务器的 CPU 应用，我们可以针对某个服务器（如网页服务器 IIS）的 CPU 使用率，做长时间的表项。一般推荐做一个月以上的长期表项才能回馈真正的 CPU 需求，然后可以进行 CPU 的平均值计算。下面就是详细的步骤。

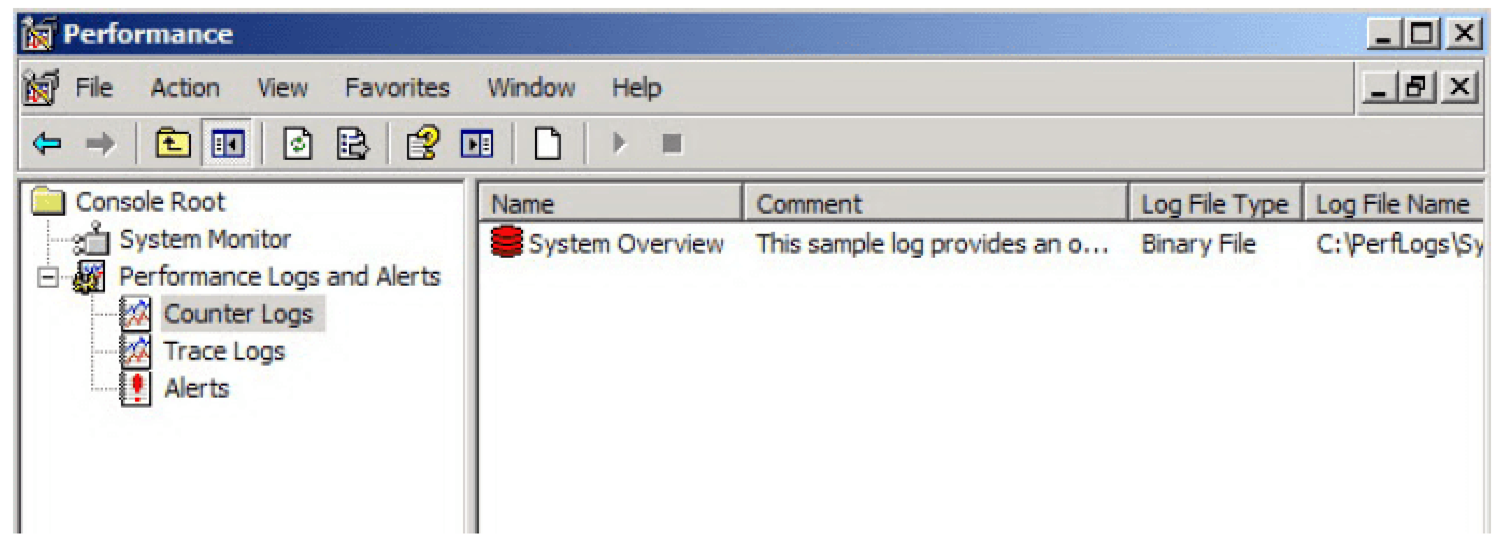
▶ 计算 CPU 的平均使用率

1. 选择 Start/Administrative Tools/Performance。如果是 Windows Server 2003，则直接选择 Start/Server Management/Performance。



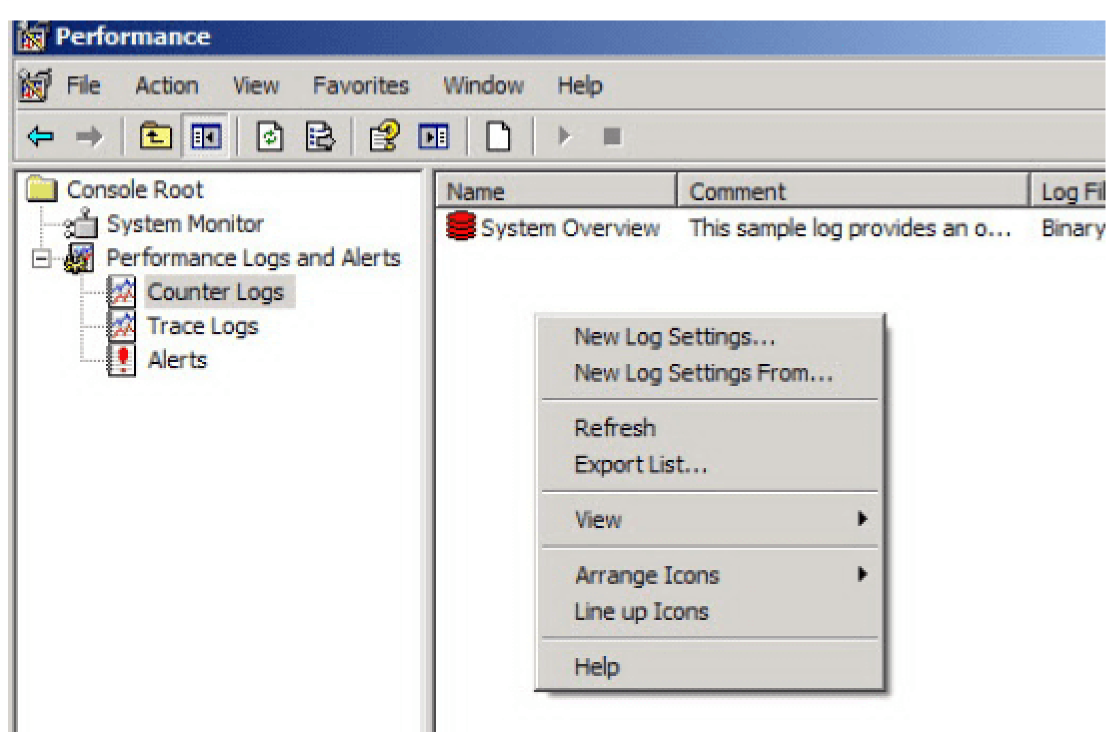
▲ 选择选项

2. 由于在 XP/2003 中已经有针对 CPU 的效能日志，我们可以直接使用。按下 Performance Logs and Alerts，选择下方的 Counter Logs，此时会在右边的窗格中弹出一个红色的 System Overview。



▲ 选择标准的系统部分

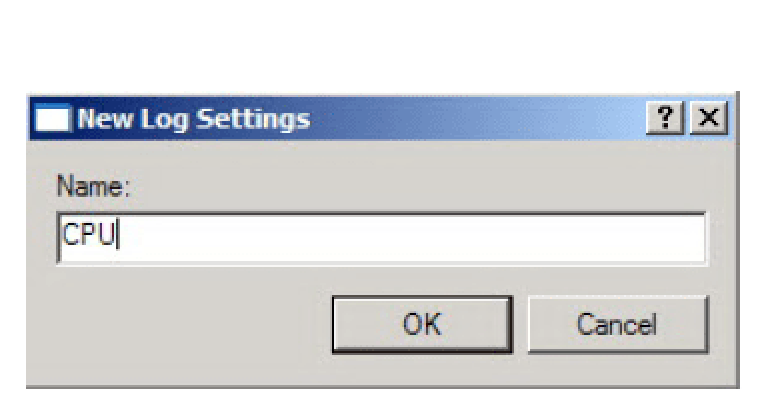
3. 我们可以自身添加一个 CPU 的计数器，直接在空白处右击键，并且选择 New Log Settings…选项。



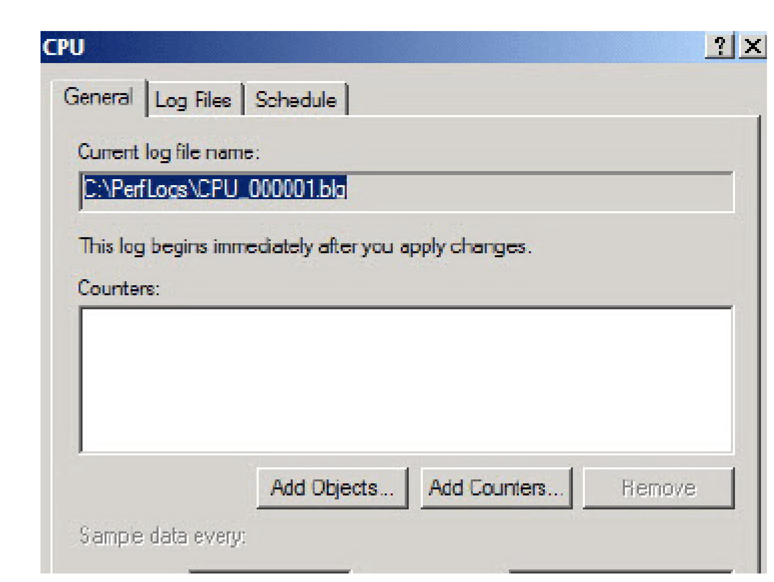
▲ 选择选项

4. 此时会弹出如下左图所示的窗口，键入 CPU，并且单击 OK 按钮。

5. 弹出如下右图所示的窗口，我们要先添加物件或计数器。在 XP 或 Windows Server 2003 中，用来监视的条目很多，我们以 CPU 为主。单击 Add Objects…按钮。

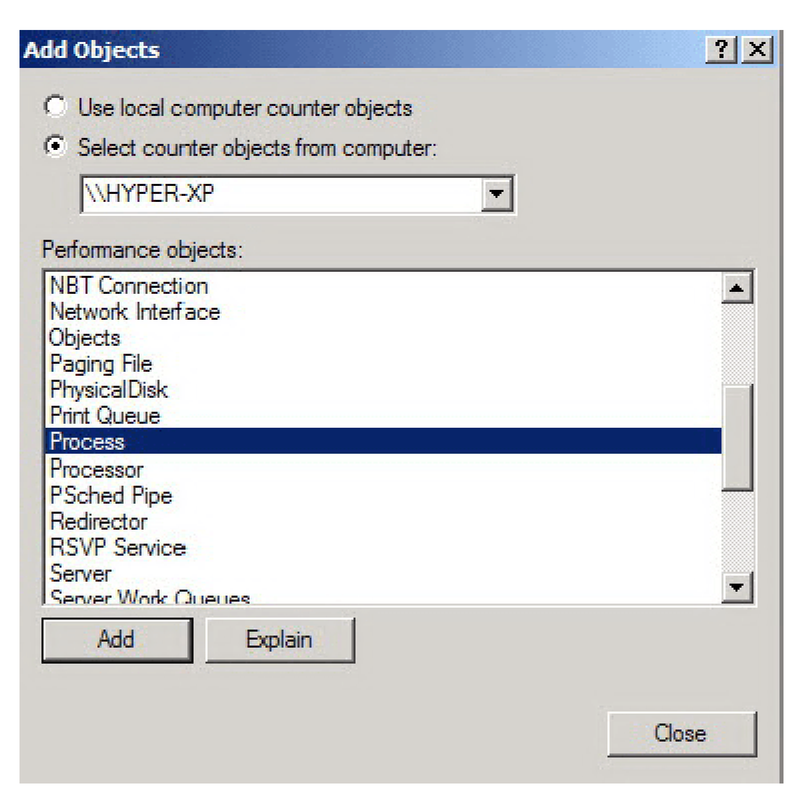


▲ 键入 CPU，这是以后计数器的名称



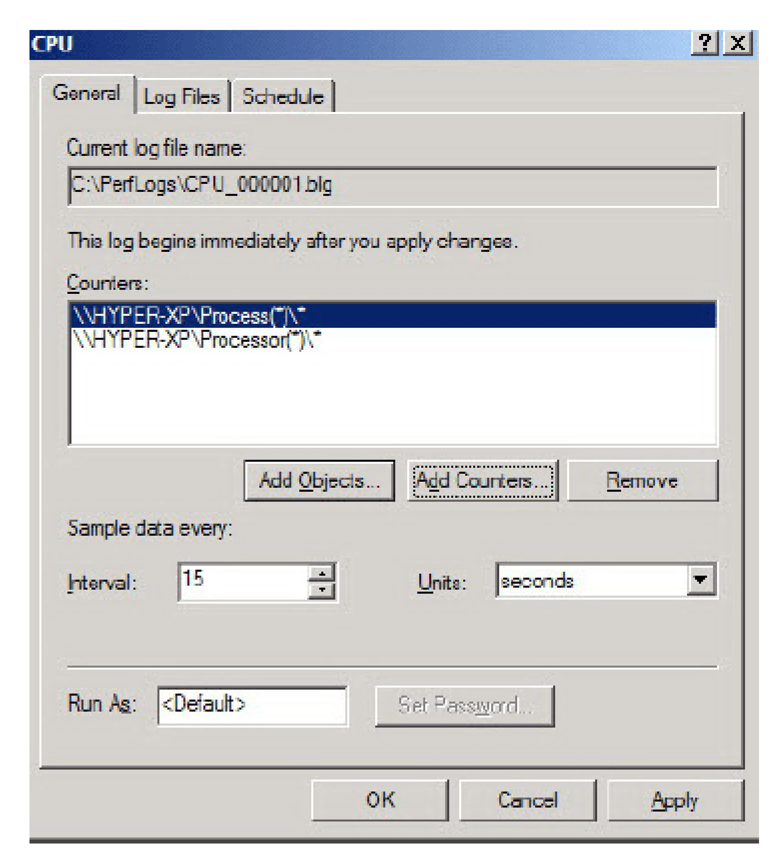
▲ 需要添加物件，就是要系统监视的对象

6. 在下图中，在 Performance objects 列表框中选择 Processors 选项，可以计算每一个 CPU 的负载。如果想看整机的 CPU 平均负载，也可以选择 Processor 选项。然后单击 Add 按钮，并且在所有的条目选择完之后单击 Close 按钮。



▲ 在此新建对象，图为 CPU 对象

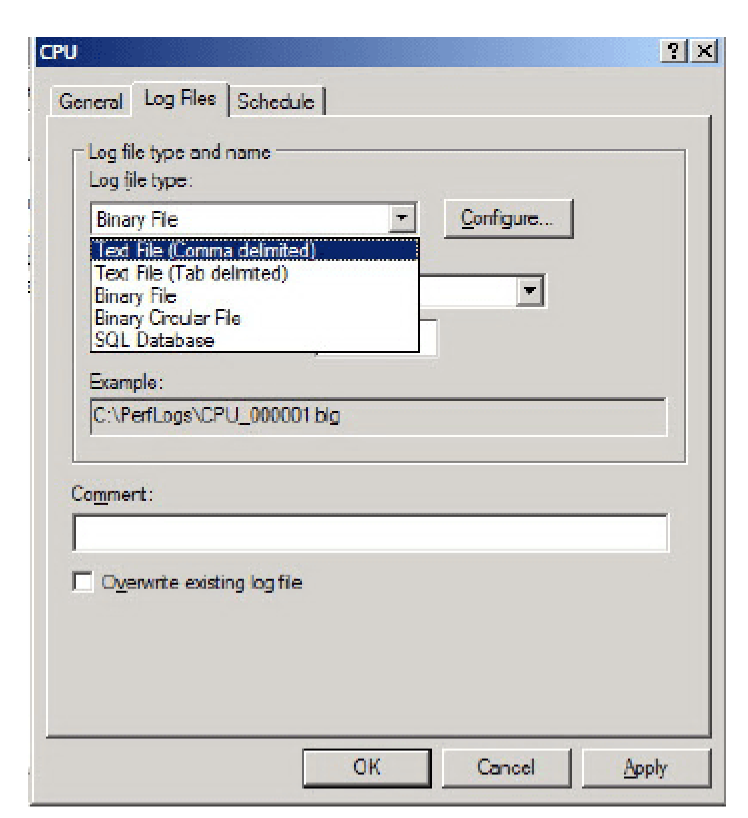
7. 之后会回到下图所示的窗口，选择的取样标的已在窗口中弹出了。我们可以在这里配置多久取样一次，一般使用默认值的 15 秒即可。原则是你取样的间隔越短，取样的数字就会越多，结果也较精确。



▲ 必须配置取样频率，也就是时间的间隔

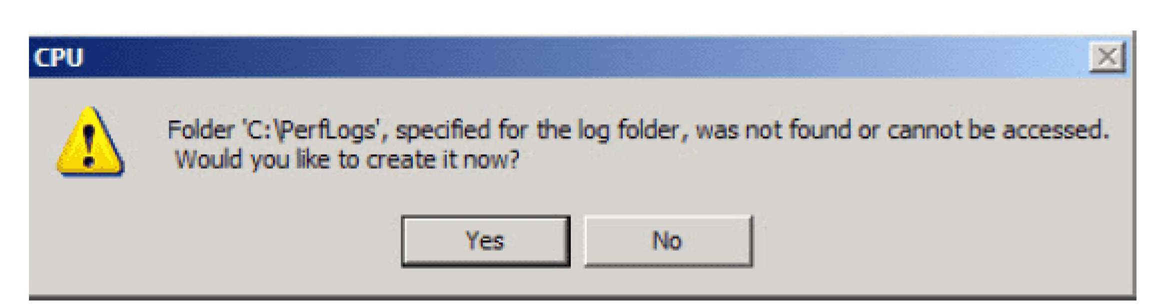
8. 此时选择 Log Files 的选项卡，然后在 Log file type 中选择 Text File（Comma delimited）

选项。如果选择一般的二进制文件，只能用 MMC 打开，无法计算长时间的平均值。另外也要记住这个文件的存放位置及文件名。之后单击 OK 按钮。



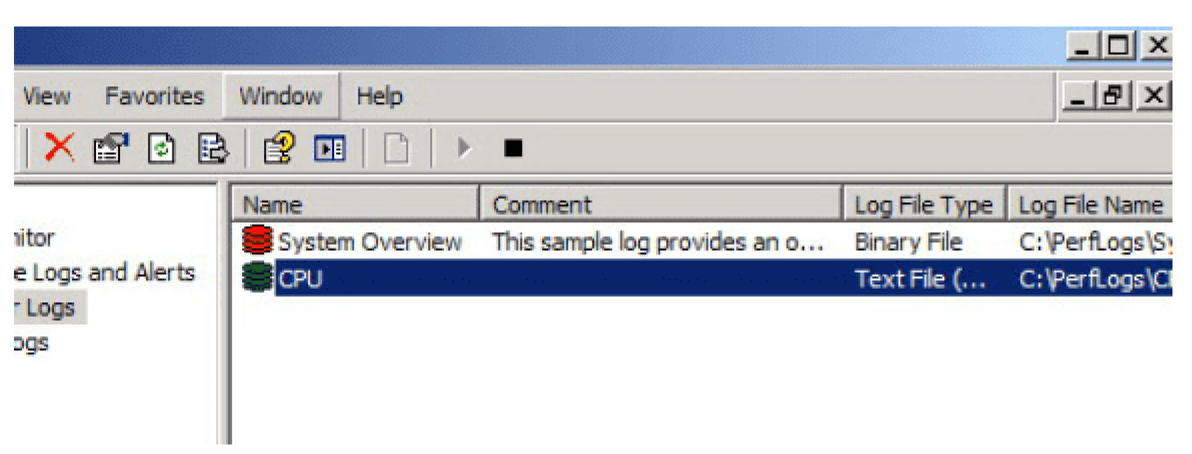
▲ 在这个选项卡的地方来选择文件格式

9. 第一次使用会弹出创建目录，如下图的报警，直接单击 Yes 按钮即可创建目录。



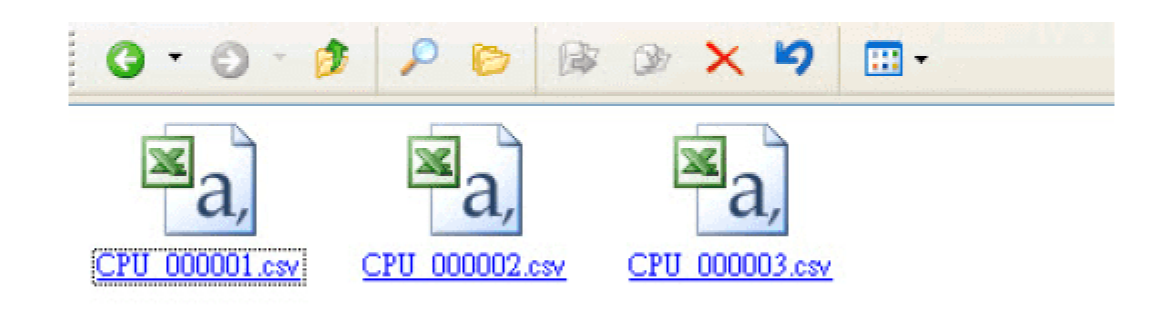
▲ 第一次用会有创建文件目录的报警

10. 此时你会发现 CPU 的图示由红色变成绿色，意味着这个表项开始运行了，而上方原来的进行图示也变灰，停止图示则变黑可以使用了。



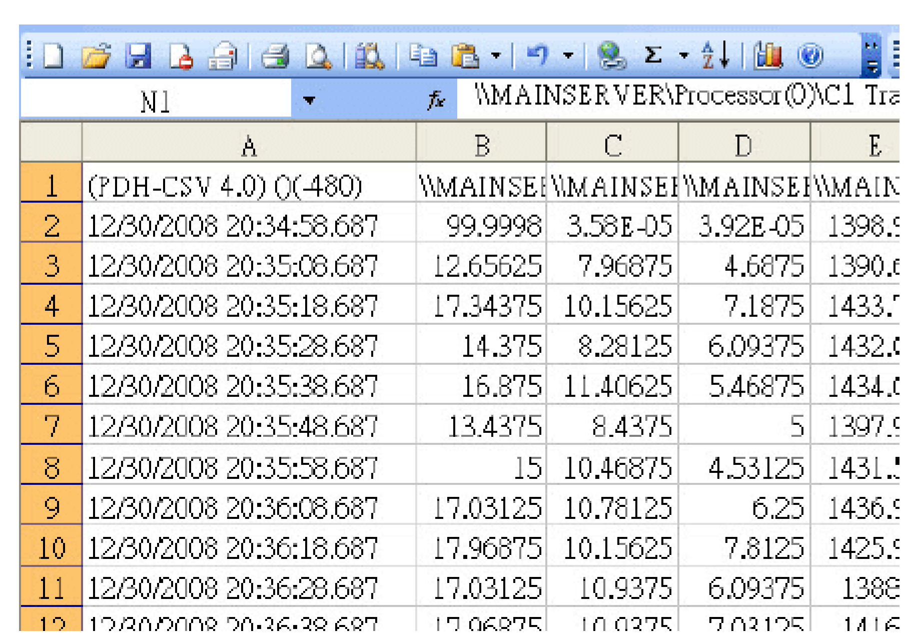
▲ 开始表项了

11. 当取样开始之后，系统即开始表项，并且将数据可写日志案中。当表项时间到达，我们使用 Excel 打开这个日志。



▲ 一般使用 Excel 打开 Csv 格式的文件

12. 此时会有很多笔表项，如下图所示，其中第一项就是 CPU 的负载值。我们要取得平均值，首先到达表项的最后一笔。



N1 | \\MAINSERVER\Processor(0)\C1 Tra

	A	B	C	D	E
1	(PDH-CSV 4.0) ()(-480)	\\MAINSEI	\\MAINSEI	\\MAINSEI	\\MAIN
2	12/30/2008 20:34:58.687	99.9998	3.58E-05	3.92E-05	1398.
3	12/30/2008 20:35:08.687	12.65625	7.96875	4.6875	1390.
4	12/30/2008 20:35:18.687	17.34375	10.15625	7.1875	1433.
5	12/30/2008 20:35:28.687	14.375	8.28125	6.09375	1432.
6	12/30/2008 20:35:38.687	16.875	11.40625	5.46875	1434.
7	12/30/2008 20:35:48.687	13.4375	8.4375	5	1397.
8	12/30/2008 20:35:58.687	15	10.46875	4.53125	1431.
9	12/30/2008 20:36:08.687	17.03125	10.78125	6.25	1436.
10	12/30/2008 20:36:18.687	17.96875	10.15625	7.8125	1425.
11	12/30/2008 20:36:28.687	17.03125	10.9375	6.09375	1388
12	12/30/2008 20:36:38.687	17.96875	10.9375	7.03125	1416

▲ 可能会有上万笔数值

13. 选择 Excel 中的平均值（下图为 Excel 2003），此时会弹出平均值，直接单击 Enter 按钮之后，即会弹出表项时间的 CPU 平均值为 6.50，即 6.5%左右。

B4236 | =AVERAGE(B2:B4235)

	A	B
4222	12/31/2008 08:18:18.765	4.375
4223	12/31/2008 08:18:28.765	3.4375
4224	12/31/2008 08:18:38.765	4.6875
4225	12/31/2008 08:18:48.765	5.15625
4226	12/31/2008 08:18:58.765	3.59375
4227	12/31/2008 08:19:08.765	11.25
4228	12/31/2008 08:19:18.765	3.90625
4229	12/31/2008 08:19:28.765	3.90625
4230	12/31/2008 08:19:38.765	5.9375
4231	12/31/2008 08:19:48.765	26.875
4232	12/31/2008 08:19:58.765	25.15625
4233	12/31/2008 08:20:08.765	20.78125
4234	12/31/2008 08:20:18.765	13.90625
4235	12/31/2008 08:20:28.765	10.9375
4236		6.501194777
4237		

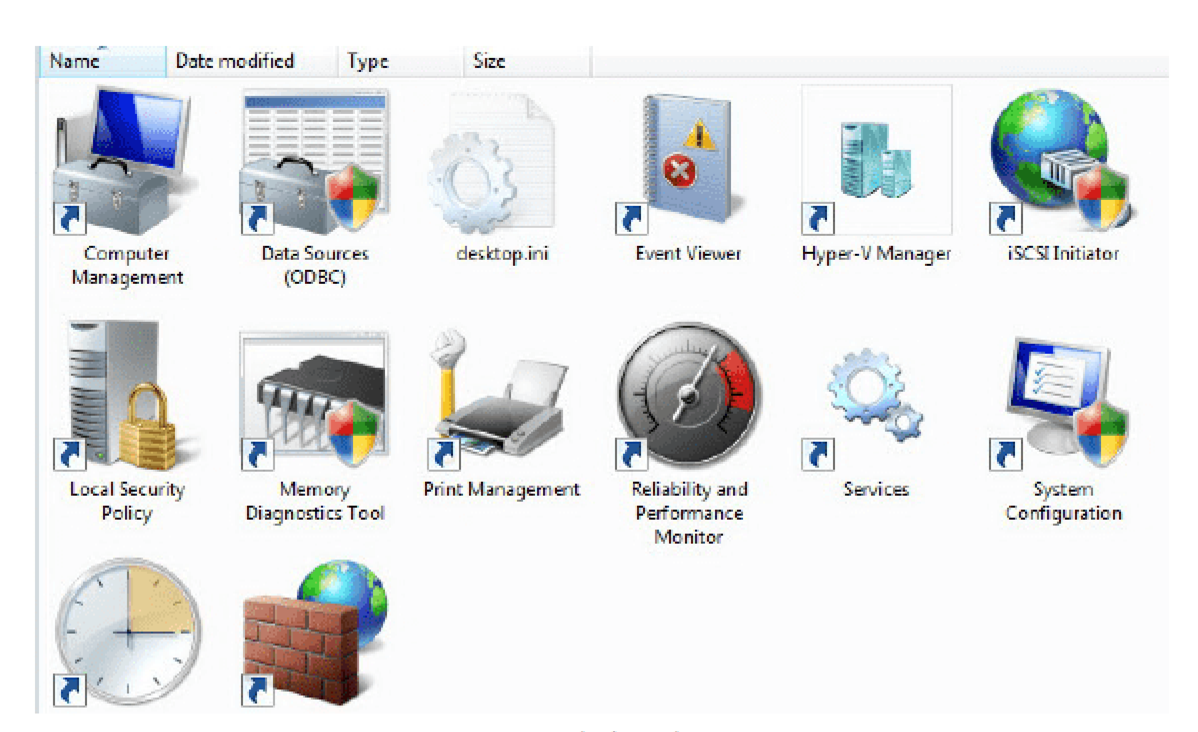
▲ 计算的平均值为 6.5%，可见一台计算机的 CPU 使用率有多低！

2. 使用 Windows Vista/7 和 Windows Server 2008

Windows Vista/7 和 Windows Server 2008 使用不同的系统效能监控方式，但大体上还是以相同的方式来处理日志。

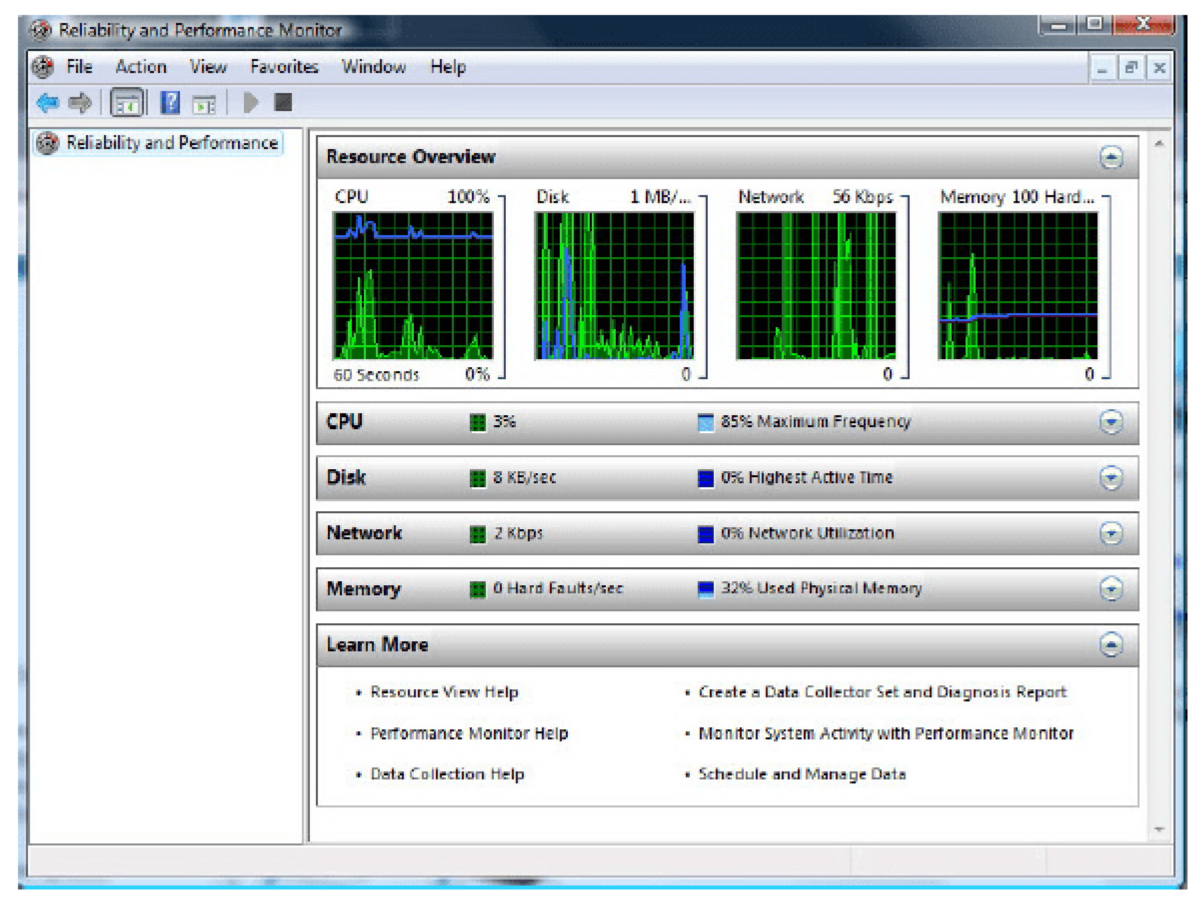
▶ Windows Vista/7 和 Windows Server 2008 计算 CPU 的平均使用率

1. 在 Windows Vista/7 下选择 Start/Control Panel/Administrative Tools/Reliability and Performance Monitor。如果是 Windows Server 2008，则直接选择“Start/Administrative Tools/Performance。



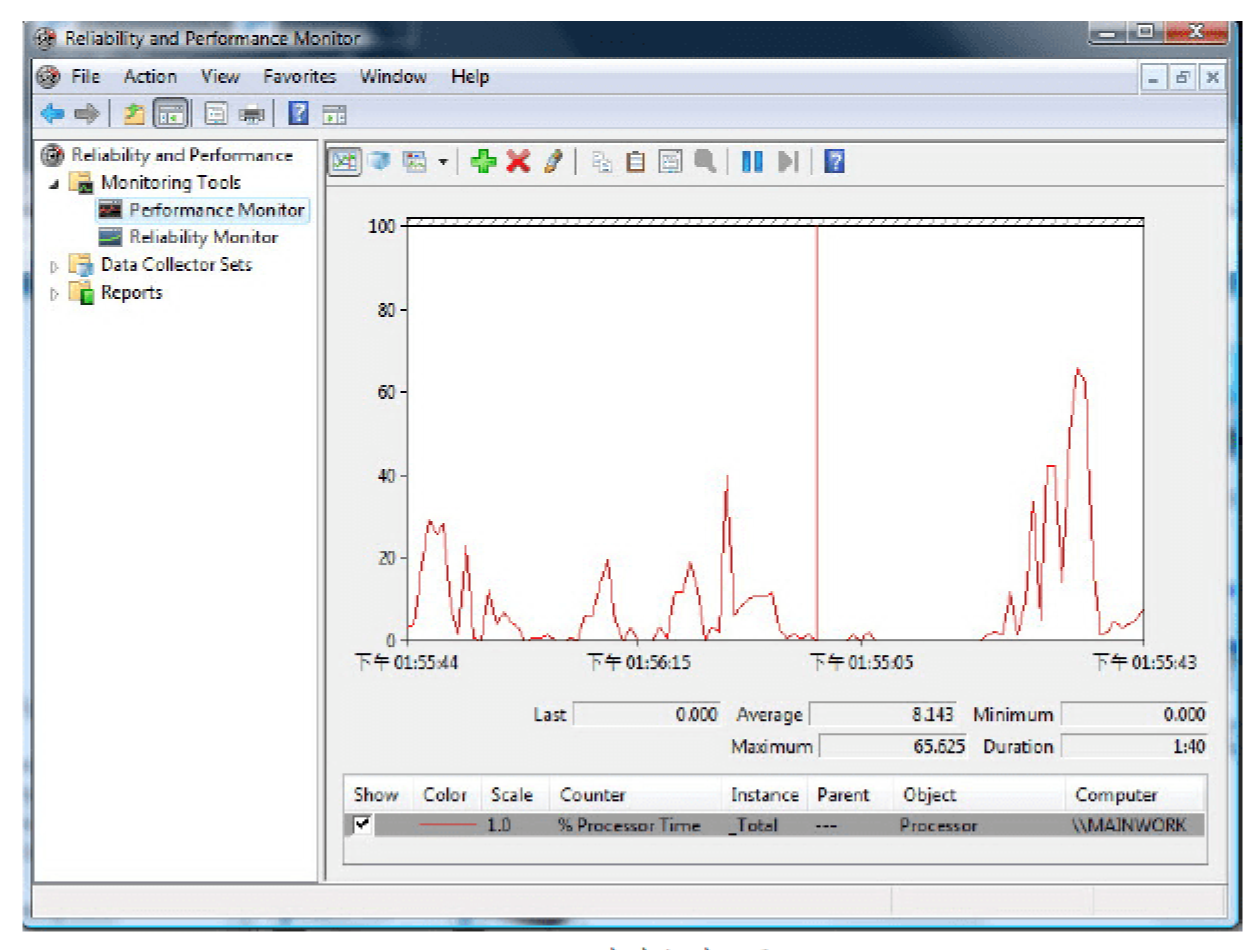
▲ 选择选项

2. 此时会弹出如下图所示的窗口，这是 Windows Vista/7 和 Windows Server 2008 的性能监视器。我们要的结果并不是当前的 CPU 效能而是一个长时间的平均。



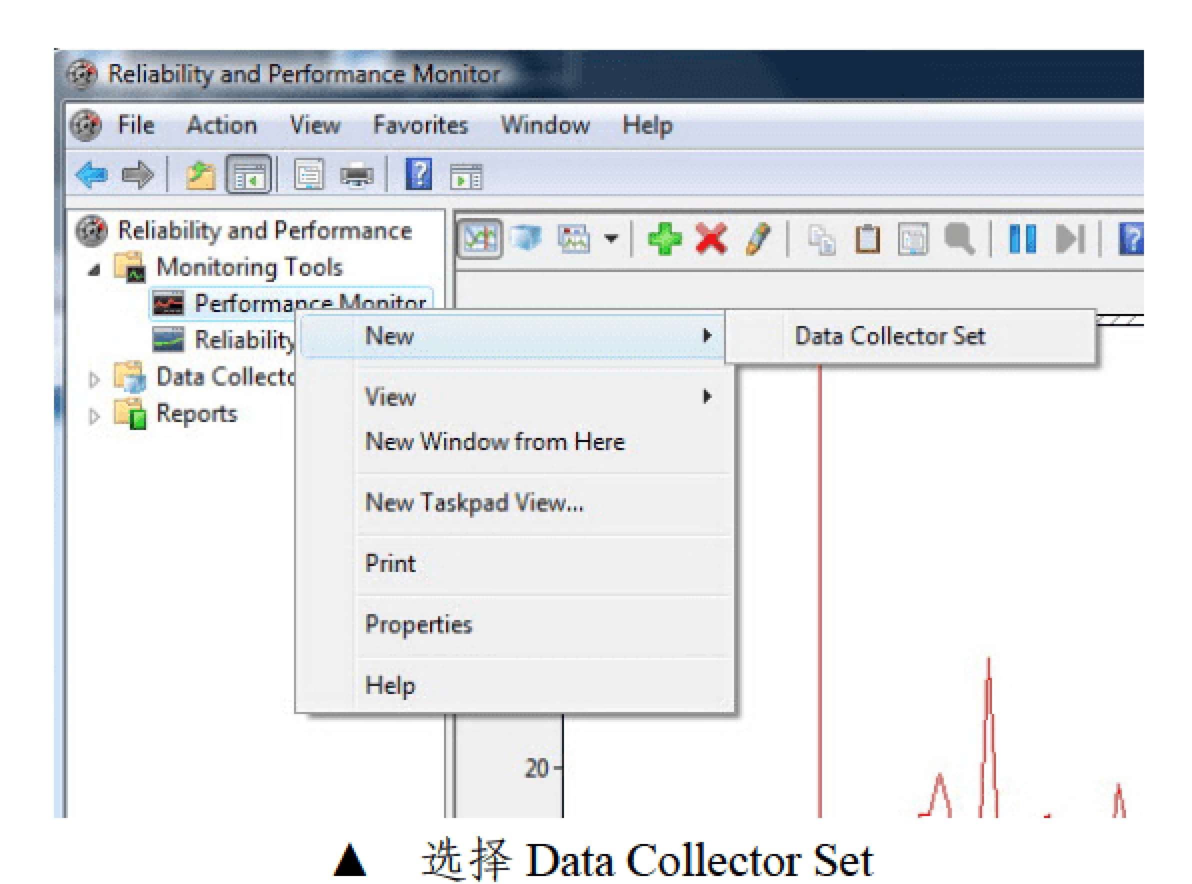
▲ 这是 Windows Vista/7 和 Windows Server 2008 的系统性能监视器

3. 选择 Monitoring Tools 中的 Performance Monitor，其中已经有默认 CPU 的效能监视了。



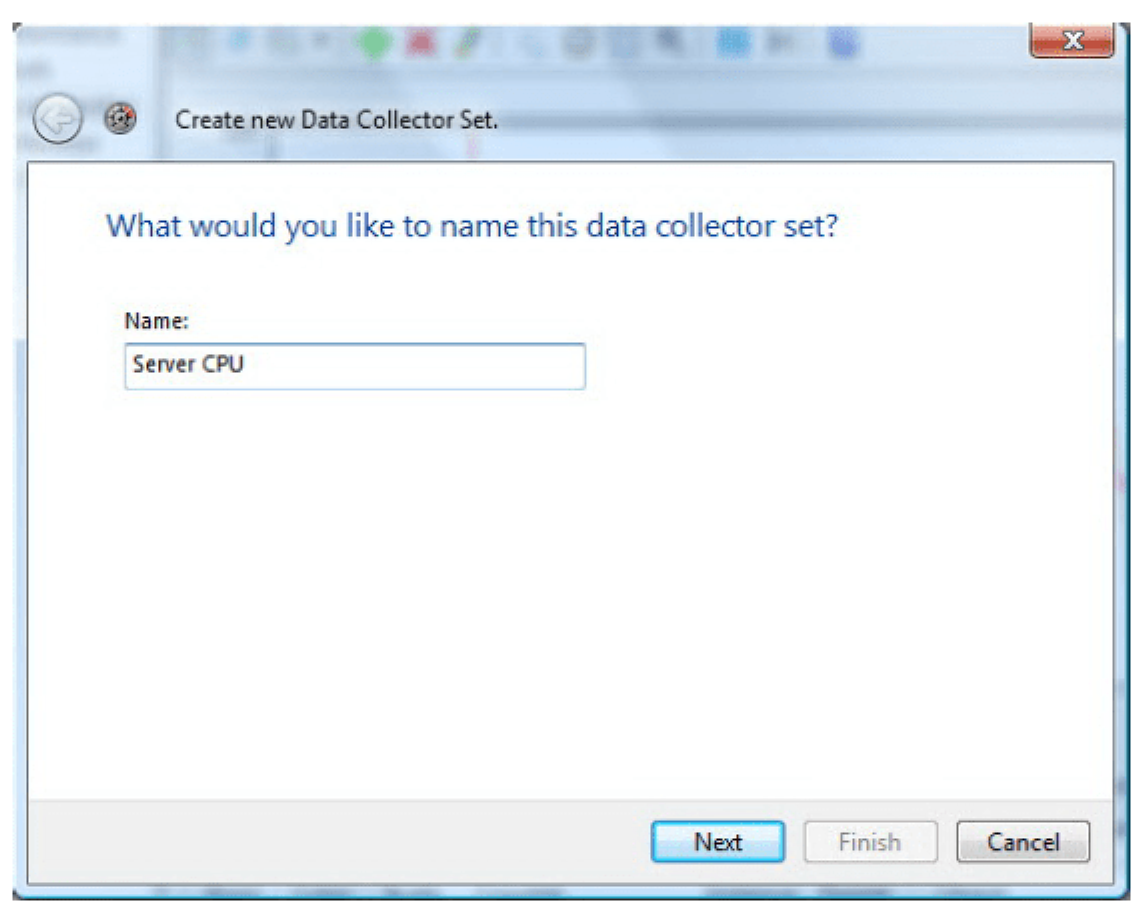
▲ 选择选项

4. 在 Windows Vista/7 和 Windows Server 2008 下是使用 Data Collector Set 来收集长时间的监视数据。我们在 Performance Monitor 的地方右击，选择 New/Data Collector Set。



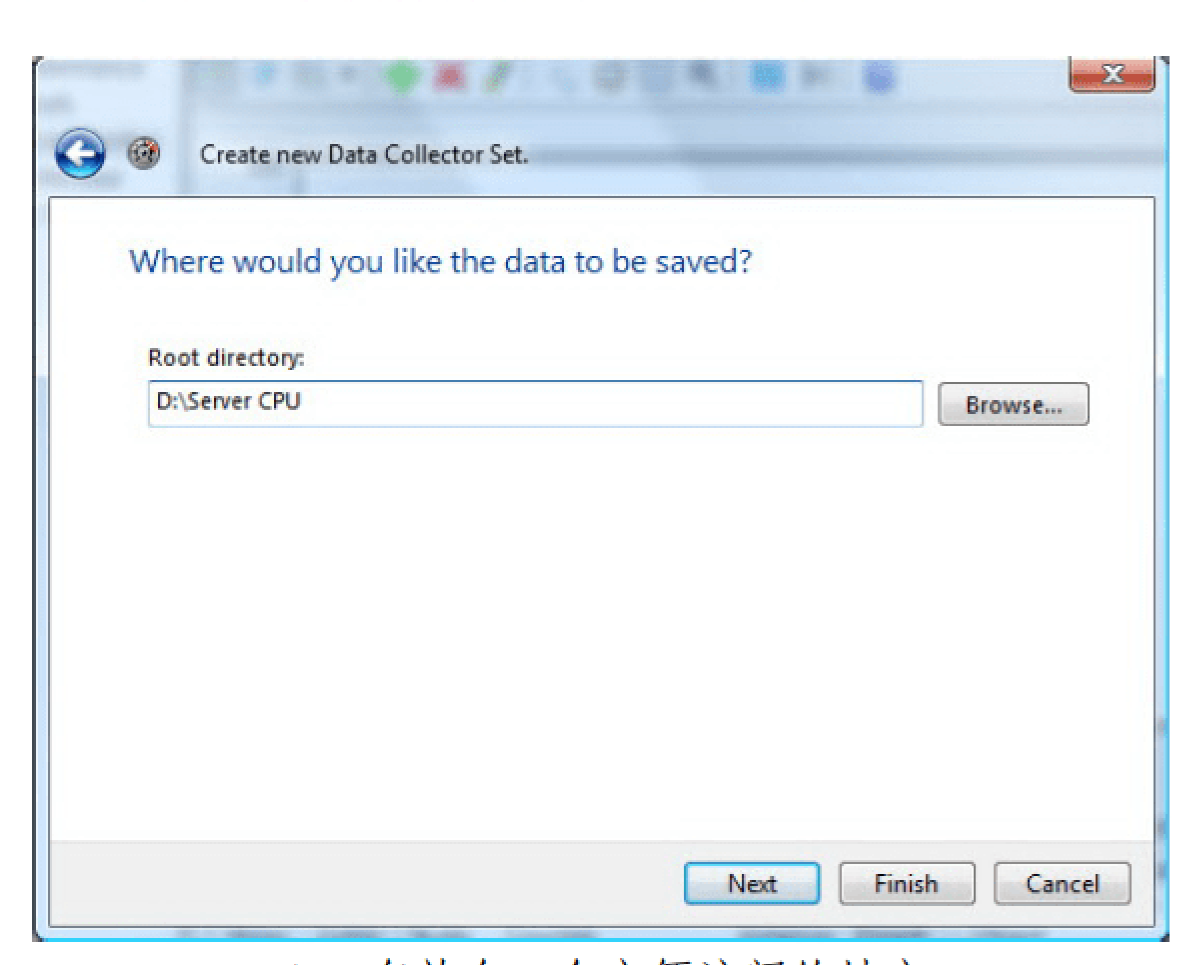
▲ 选择 Data Collector Set

5. 我们替这个数据收集器命名为 Server CPU 之后单击 Next 按钮。



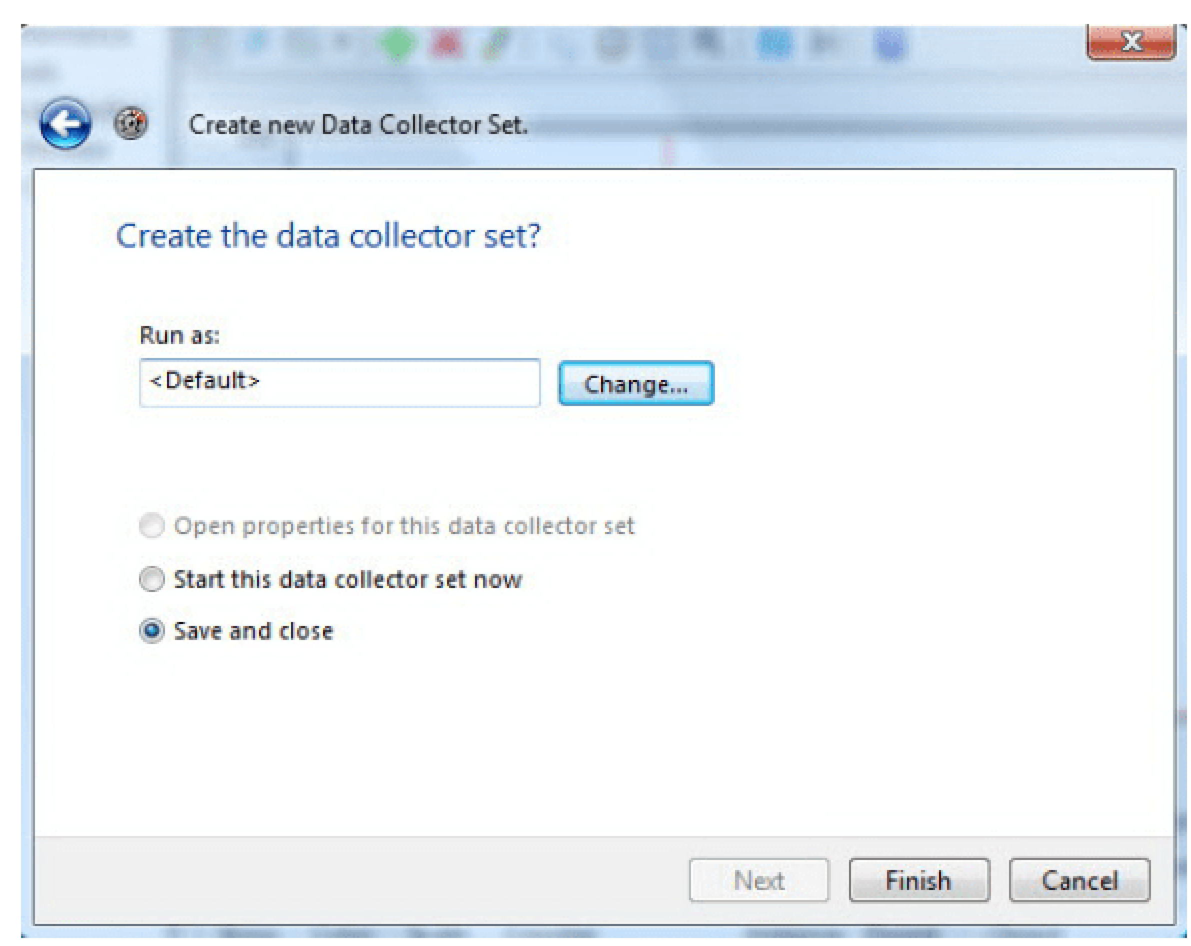
▲ 一样要命名

6. 我们将这个日志放在一个较容易访问的地方，如 D:\Server CPU，之后单击 Next 按钮。



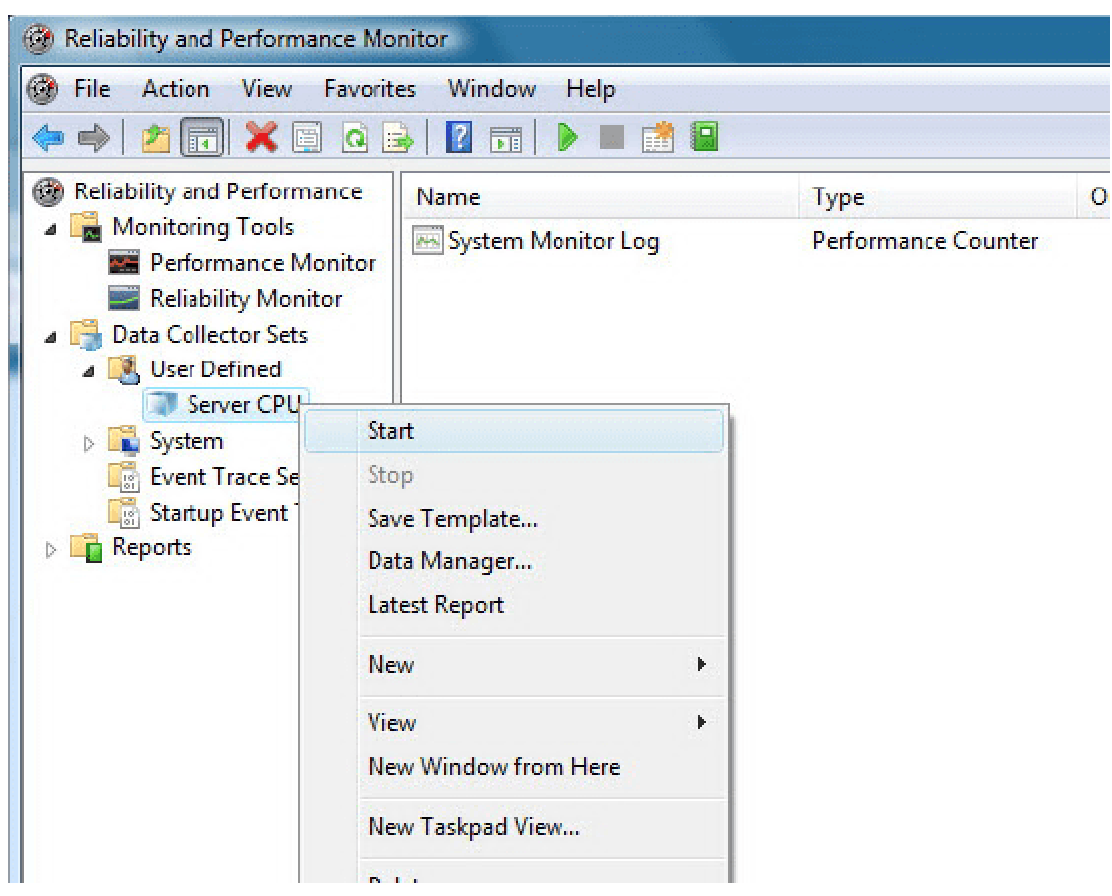
▲ 存放在一个方便访问的地方

7. 接下来是运行的权限，不需要更改直接单击 Finish 按钮即可。



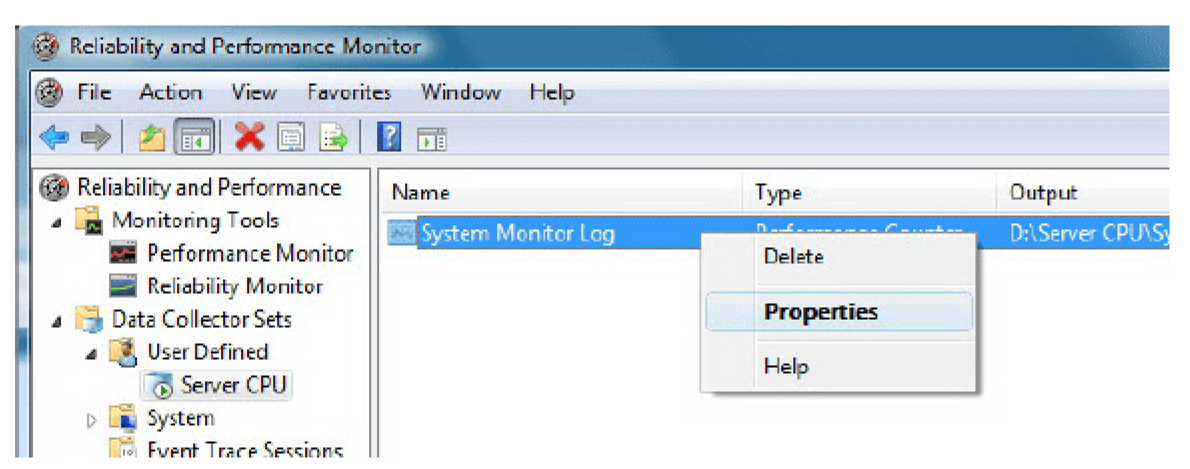
▲ 不需要管权限，直接结束即可

8. 在创建之后，在 Data Collector Set 的地方会弹出刚才键入的 Server CPU。



▲ 这里是之前创建的

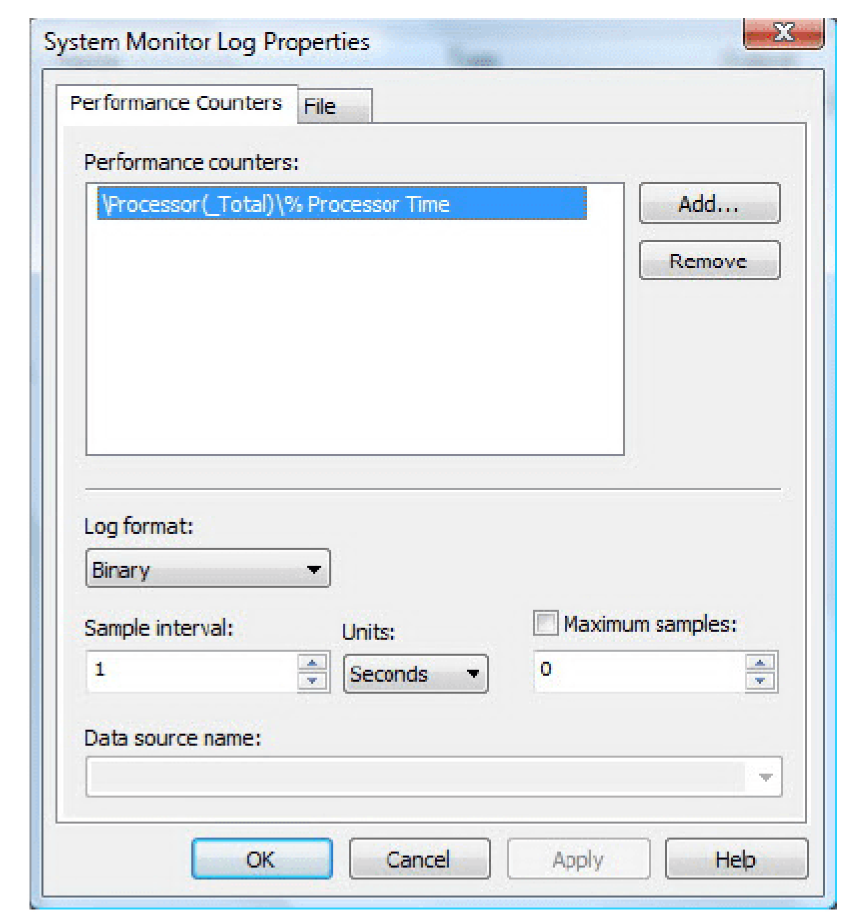
9. 我们在下图中右边的窗格看到 System Monitor Log，这就是配置文件。也是在其上右击，并且选择 Properties。



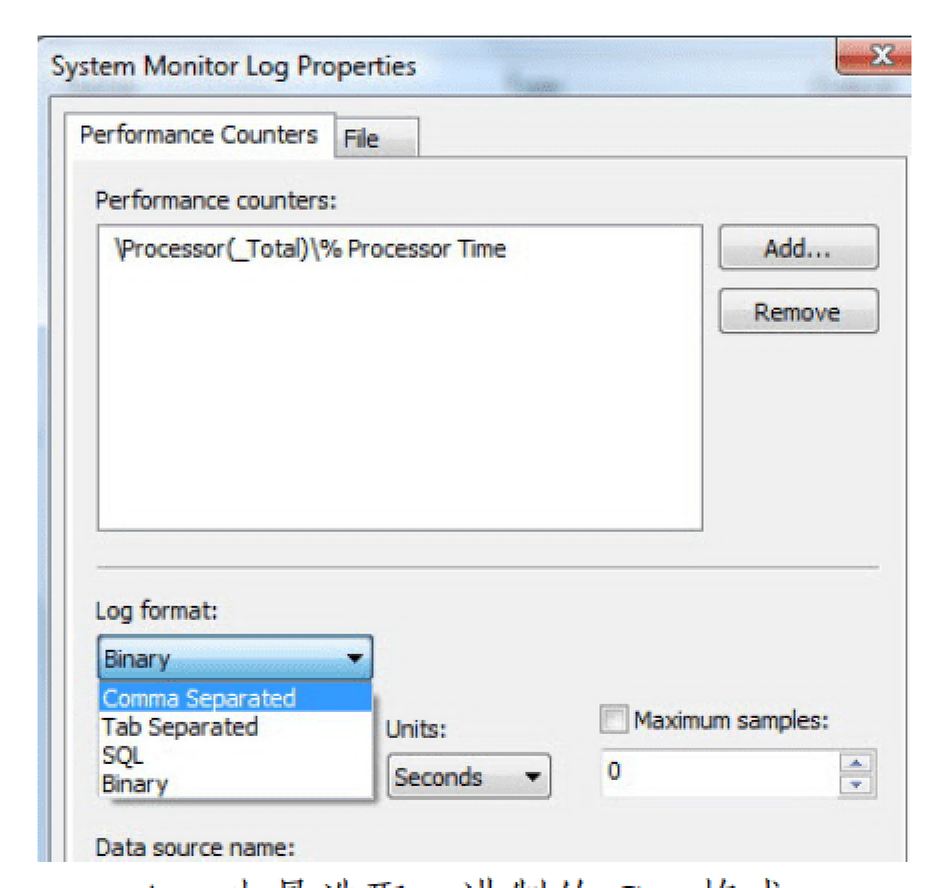
▲ 也要创建一个系统文件

10. 此时的画面和 XP/和 Windows Server 2003 的大同小异，就是加入不同的对象。由于这边会自动将 CPU 加入，因此不需要再加了。

11. 也不要忘了将格式存放目录配置完毕。

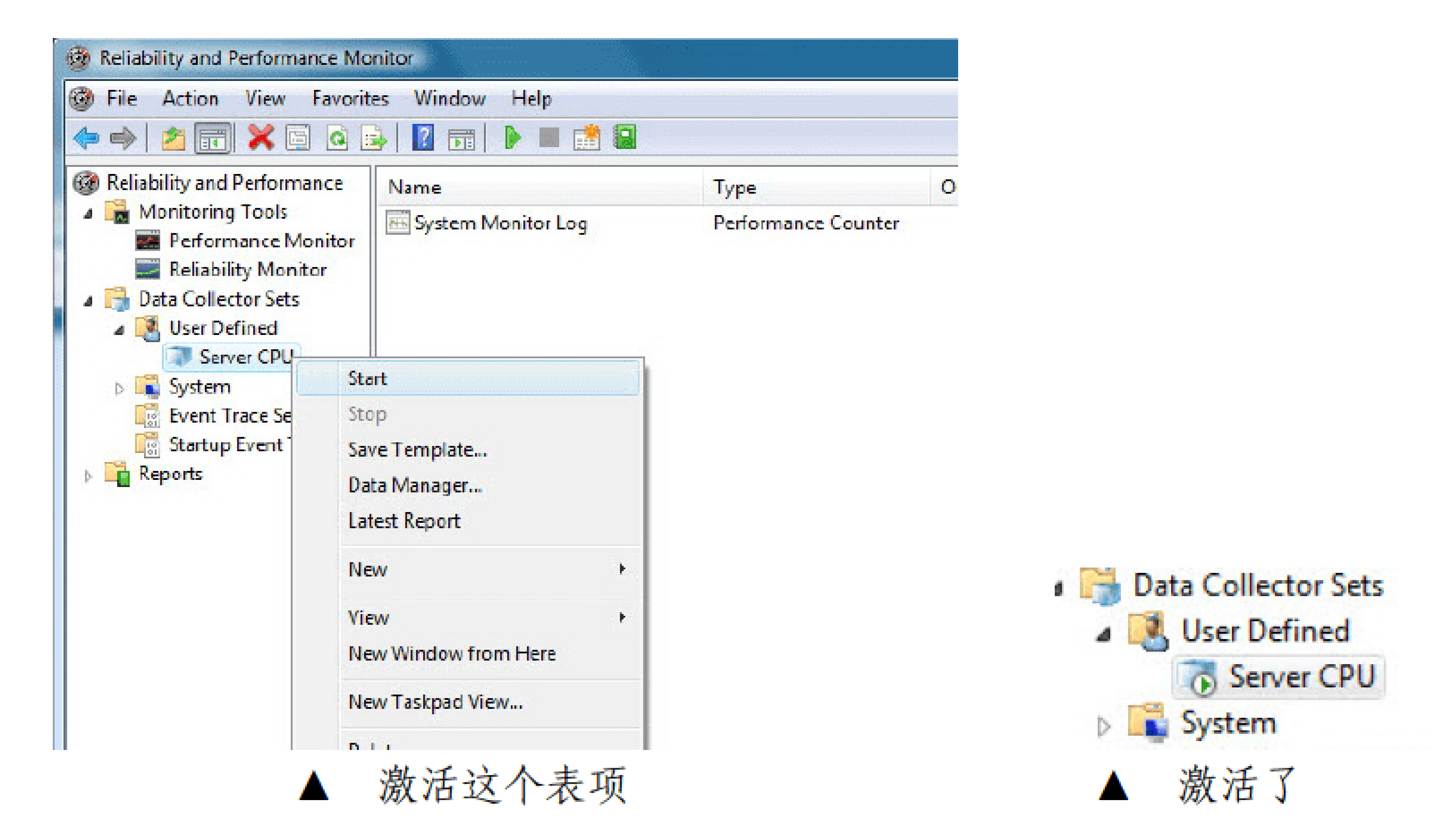


▲ 在 Windows Vista/7/和 Windows Server2008 中已经创建好了



▲ 也是选取二进制的 Csv 格式

12. 在其上右击，并且选择 Start，整个监视过程就会激活了。此时会弹出一个激活后的绿色标识。



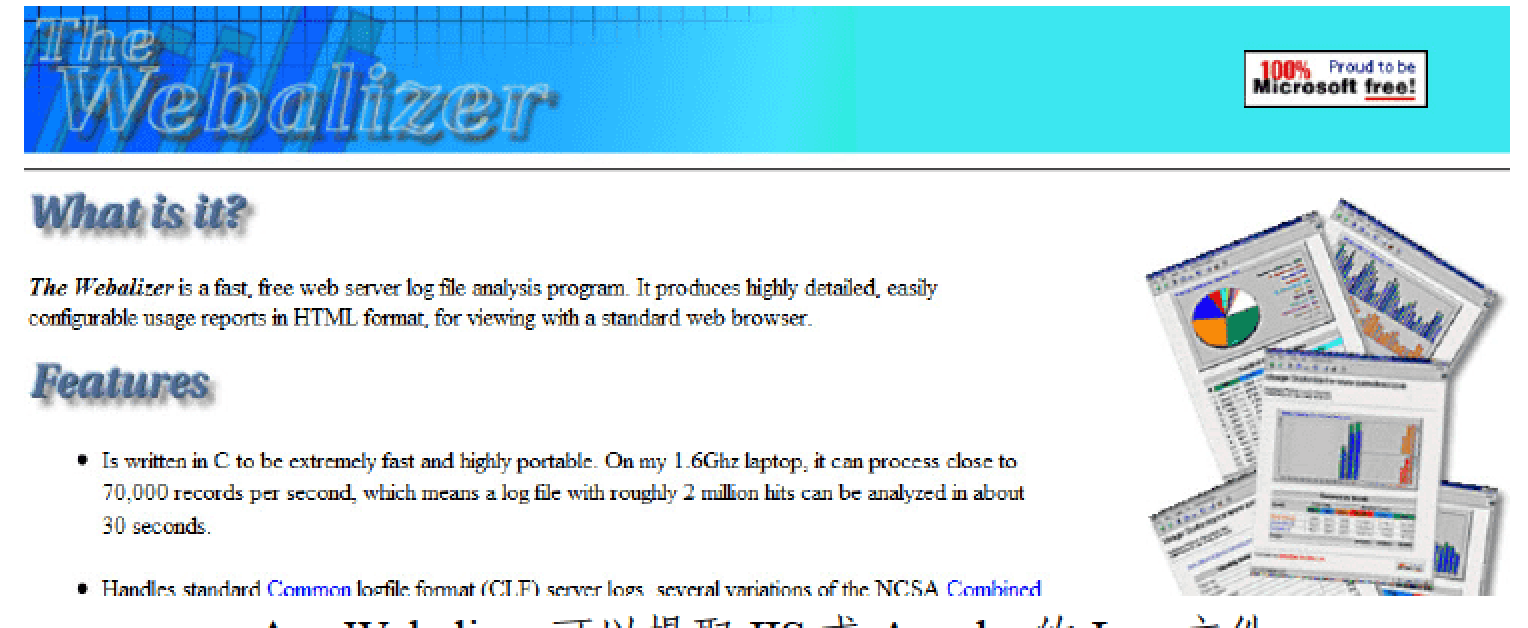
▲ 激活这个表项　　▲ 激活了

之后我们也就可以利用 Excel 打开来计算其 CPU 的使用平均值了。

6.1.2 专业提取工具 PAL Tool

虽然 CPU 是虚拟化中的重要指针，然而虚拟机的使用单非 CPU 可决定其平顺度或合并率的，系统的整体效率也相当重要。在 Windows 系统中，我们可以表项各种计数器或监视效能的下标，但单用 Excel 来计算平均值实在无法真正表达出系统各部分的“瓶颈”和长期的使用状况，因此我们需要一个专业的提取工具来落实这件事。

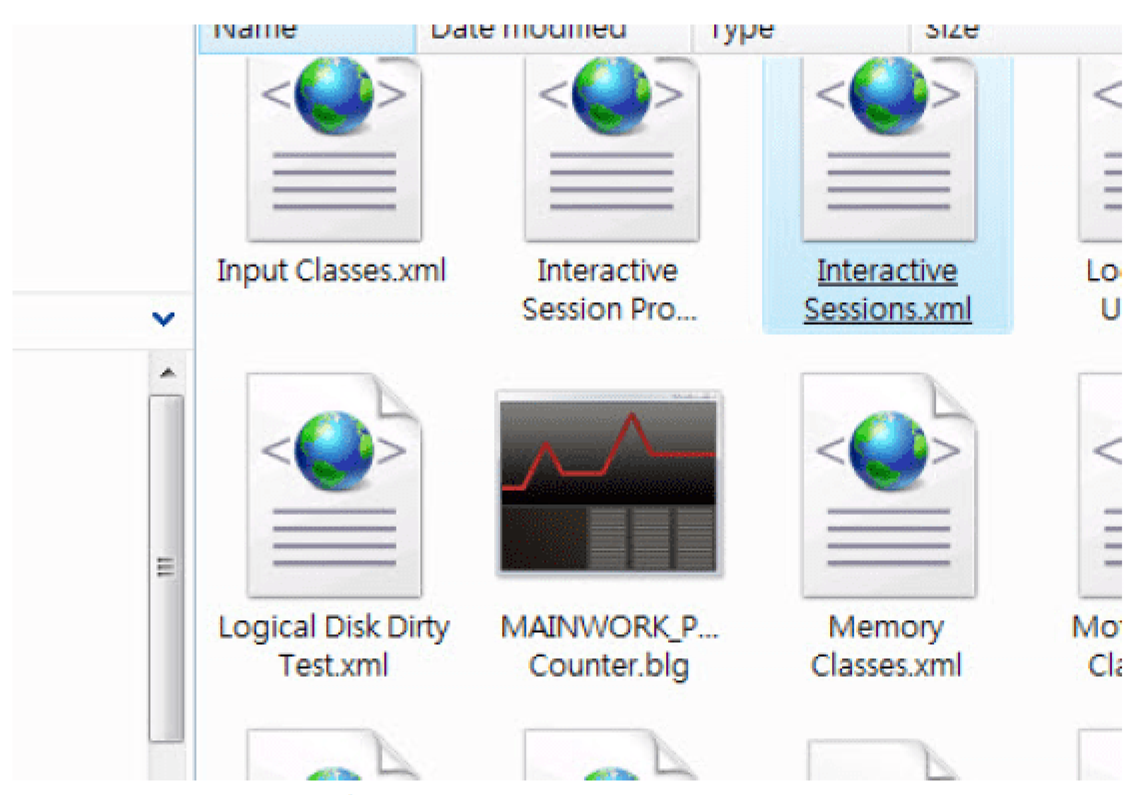
就像 Webalizer 可以提取 Apache 和 IIS 等网页的 Log 文件一样，窗口系统下也有一个第三方工具来提取 Log 文件，并且以图表和颜色的直觉方式来显示各种指针的长期平均效能，并且找出“瓶颈”，我们就来看看 PAL Tool 这个工具。



▲ Webalizer 可以提取 IIS 或 Apache 的 Log 文件

1. PAL Tool 简介

PAL Tool 和 Webalizer 一样，就是用来提取窗口操作系统的效能日志案，不管使用的是窗口默认的 blg 文件或是 CSV 等，都可以用来提取。由于窗口系统在效能上的表项相当完整及严谨，尤其是在服务器上，包括了数据库的 SQL、邮箱的 Exchange 或是管理网域的 Active Directory，这些服务是常弹出效能“瓶颈”的地方，因此在时间、错误、数值上，都能完整地被表项到 Log 文件中。



▲ 可以提取各种格式的窗口操作系统的 Log 文件

PAL Tool 在提取了 Log 文件之后，会将结果用最匹配使用习惯的方法键入，输出的结果是 Html，并且拥有不同的颜色来表示系统的严重性，如红色代表效能瓶颈，黄色代表有危险及绿色为系统平顺运行等等，并且还有不同的统计图表来显示数字的变化，我们就来看看使用的方式。

Alerts

An alert is generated if any of the above thresholds were broken during one of the time intervals analyzed. An alert condition of OK means that the counter instance was analyzed, but did not break any thresholds. The background of each of the values represents the highest priority threshold that the value broke. See the 'Thresholds Analyzed' section as the color key to determine which threshold was broken. A white background indicates that the value was not analyzed by any of the thresholds.

Time	Condition	Counter	Min	Avg	Max	Hour Tren
8/14/2007 3:40:00 PM	Average disk responsiveness is slow - more than 15ms	\\ZACH-PC\PhysicalDisk(0 C:)\Avg. Disk sec/Write	0.012	0.012	0.012	0
8/14/2007 3:41:00 PM	Average disk responsiveness is slow - more than 15ms	\\ZACH-PC\PhysicalDisk(0 C:)\Avg. Disk sec/Write	0.011	0.011	0.011	-0.077
8/14/2007 3:44:00 PM	Average disk responsiveness is slow - more than 15ms	\\ZACH-PC\PhysicalDisk(0 C:)\Avg. Disk sec/Write	0.017	0.017	0.017	0.076
8/14/2007 3:45:00 PM	Average disk responsiveness is slow - more than 15ms	\\ZACH-PC\PhysicalDisk(0 C:)\Avg. Disk sec/Write	0.016	0.016	0.016	0.049
8/14/2007 3:40:00 PM	Average disk responsiveness is slow - more than 15ms	\\ZACH-PC\PhysicalDisk(_Total)\Avg. Disk sec/Write	0.012	0.012	0.012	0
8/14/2007 3:41:00 PM	Average disk responsiveness is slow - more than 15ms	\\ZACH-PC\PhysicalDisk(_Total)\Avg. Disk sec/Write	0.011	0.011	0.011	-0.077

▲ 一般系统有点吃紧时就用黄色代替，到达瓶颈时就用红色表示

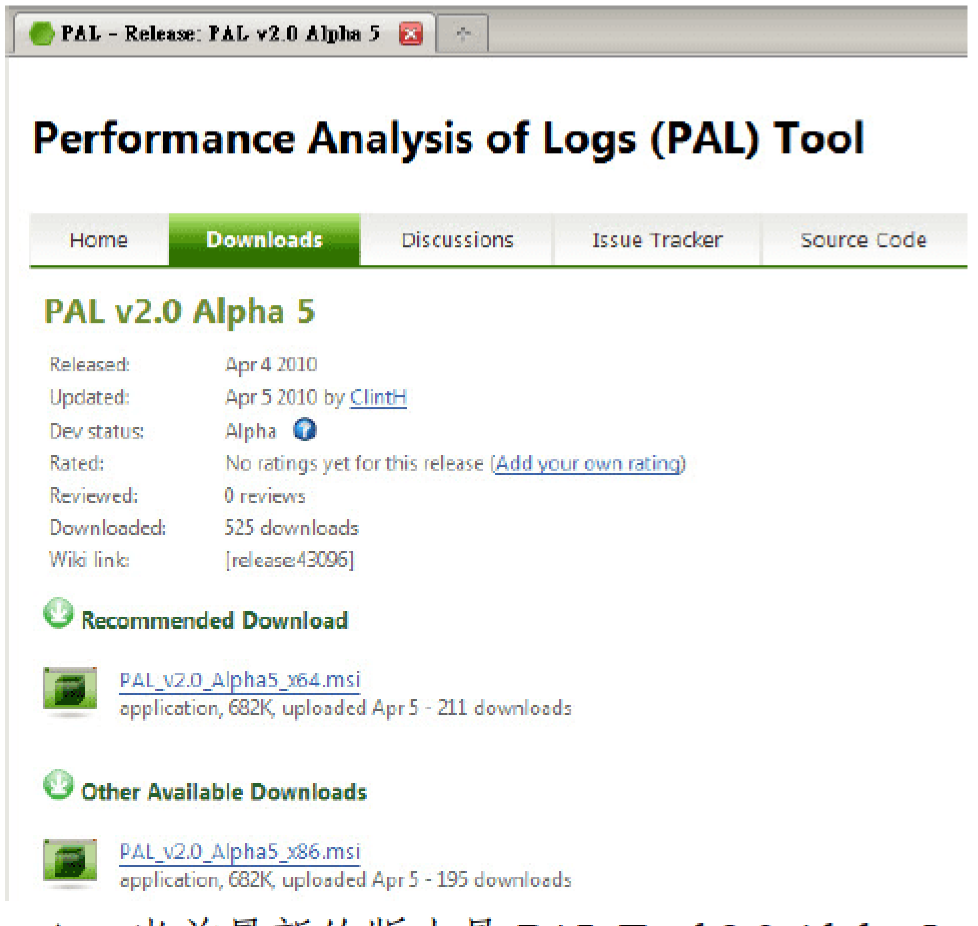
2. 下载 PAL Tool

要完整使用 PAL Tool 的功能，还必须安装微软的 Log Parser 来解析 Log 文件，以及 Office Web Components 2003 来生成标准的 Office Html 图表，还有 Chat Control 这个文件，更重要的就是 Windows PowerShell 运行环境。这五个文件的下载点分别如下。

▶ PAL Tool 以及相关文件的下载点

- PAL Tool：
  www.codeplex.com/PAL/Release/ProjectReleases.aspx?ReleaseId=16807
- Log Parser：
  http://www. microsoft.com/downloads/details.aspx?FamilyID=7287252c-402e-4f72-97a5-e0fd290d4b76&displaylang=en
- Office 2003 Web Components：
  http://www.microsoft.com/downloads/details.aspx?FamilyID=7287252c-402e-4f72-97a5-e0fd290d4b76&displaylang=en
- Chat Control：
  http://www.microsoft.com/downloads/details.aspx?FamilyID=130f7986-bf49-4fe5-9ca8-910ae6ea442c&DisplayLang=en
- PowerShell：

http://download.microsoft.com/download/d/e/7/de7cb629-73a3-49cc-b796-50dfaaf1b833/WindowsServer2003-KB926139-v2-x86-ENU.exe



▲ 当前最新的版本是 PAL Tool 2.0 Alpha 5

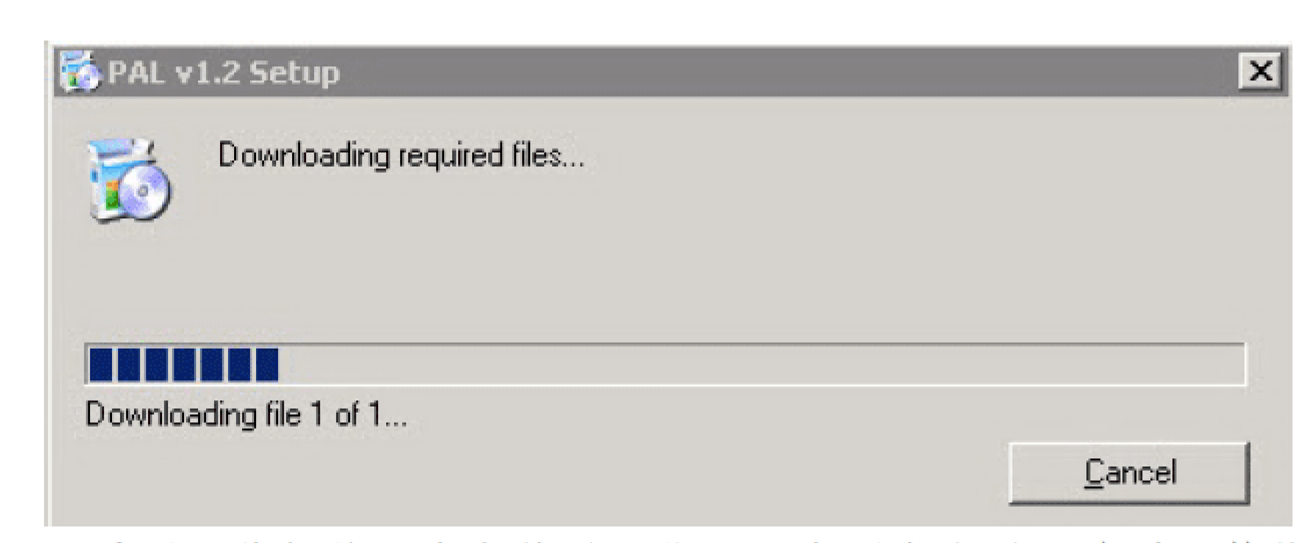
3. 安装 PAL Tool

PAL Tool 的安装十分简单，我们就来看看。

1. 先确定安装的计算机能连上网络，将下载回来的压缩文件解压，并且直接运行下载回来的 msi 文件。由于 PAL Tool 需要 Windows 中的一些部件，因此会开始下载需要的部件。

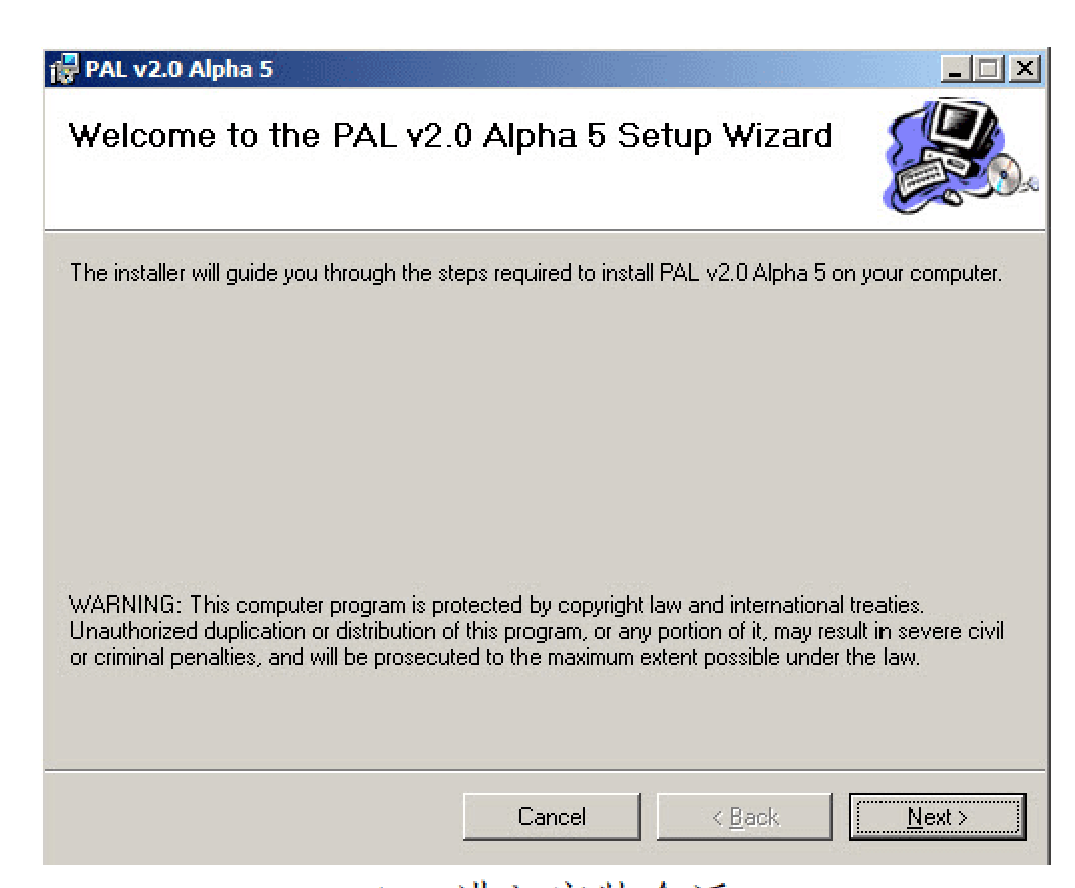


▲ 选择选项



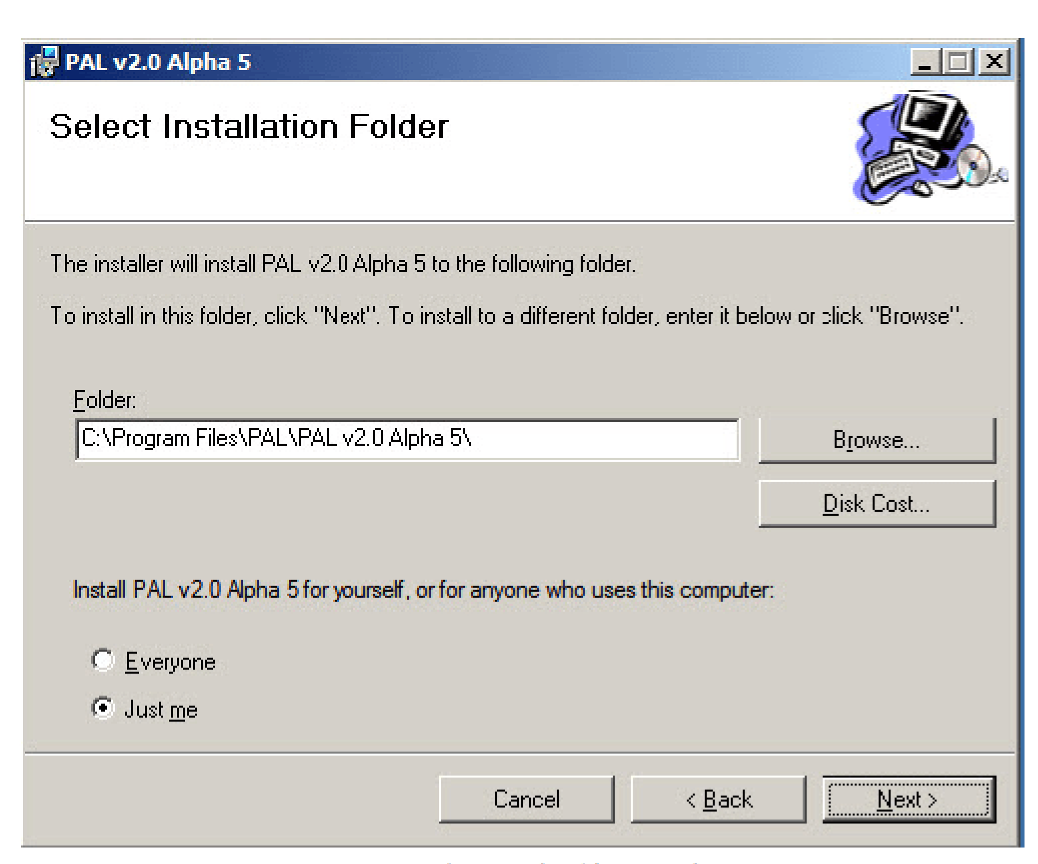
▲ 需要一些部件，在安装时下载，至少要半小时，请耐心等待

2. 下载落实之后进入安装会话，直接单击 Next 按钮。



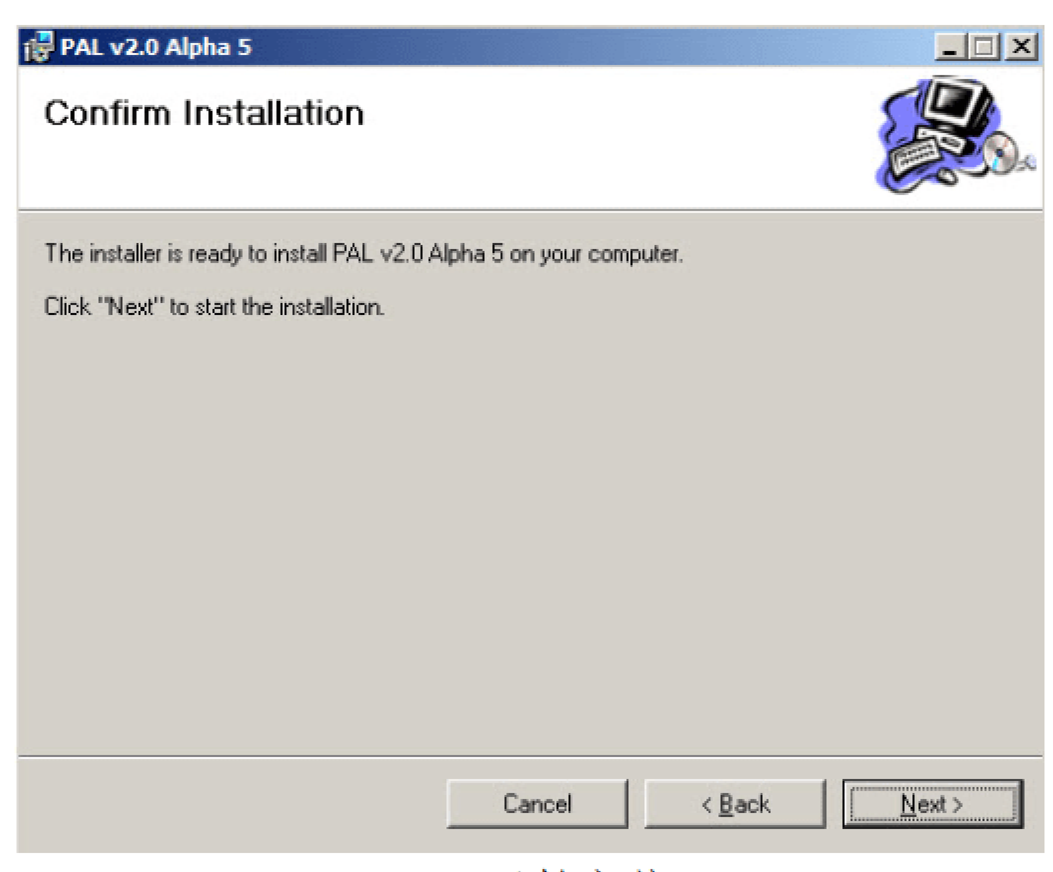
▲ 进入安装会话

3. 定义安装目录以及允许的用户，单击 Next 按钮。



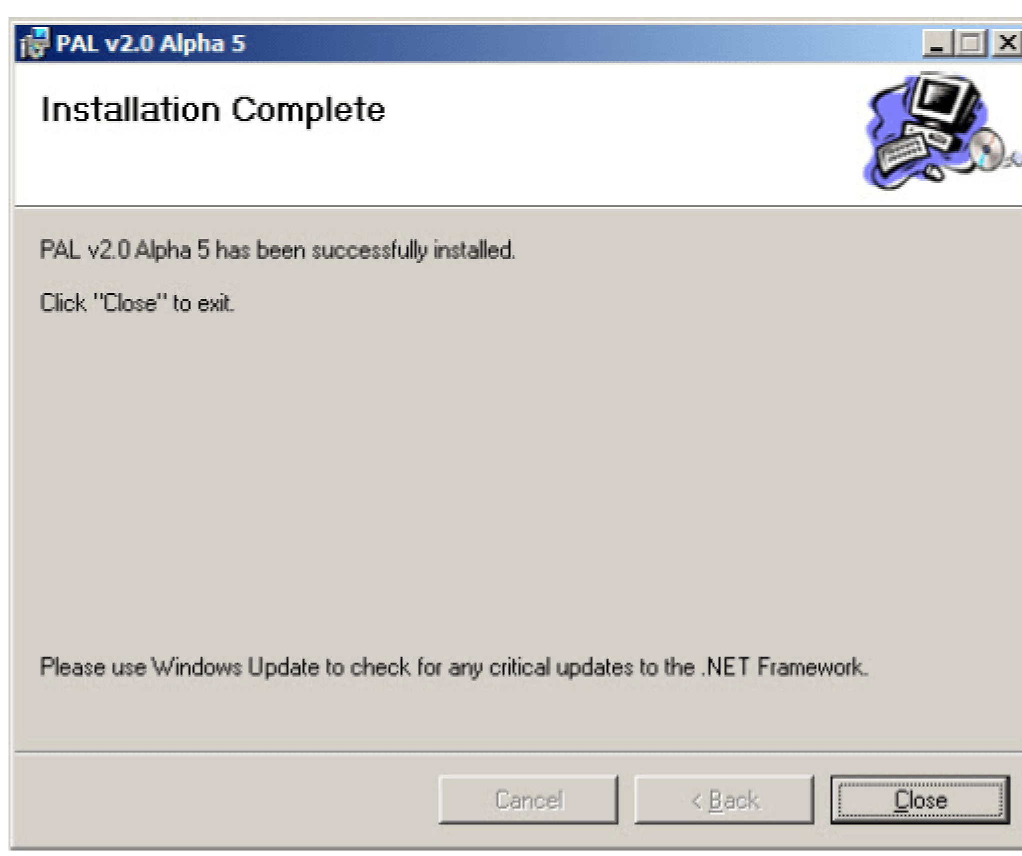
▲　定义安装目录

4. 接下来直接单击 Next 按钮即开始安装。



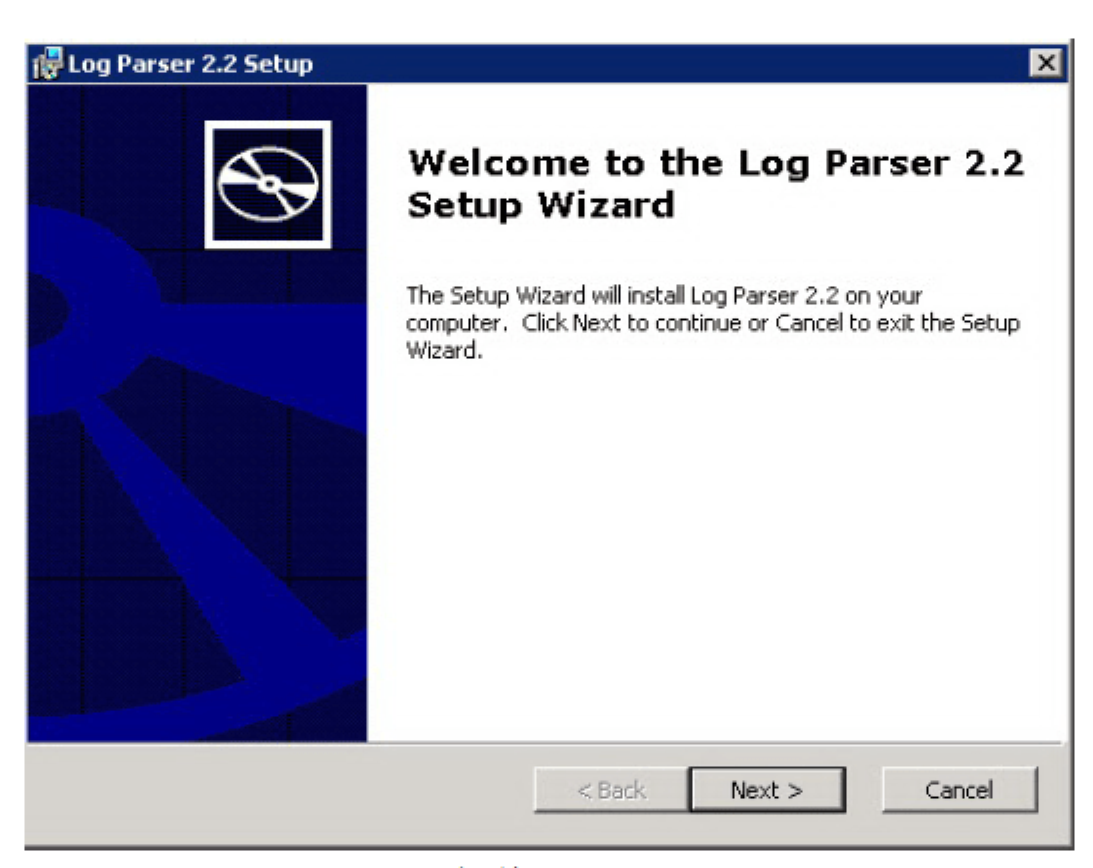
▲　开始安装

5. 安装完毕之后单击 Close 按钮结束。弹出如下图所示的窗口时意味着安装完毕。此时还没办法使用，必须安装其他构件。



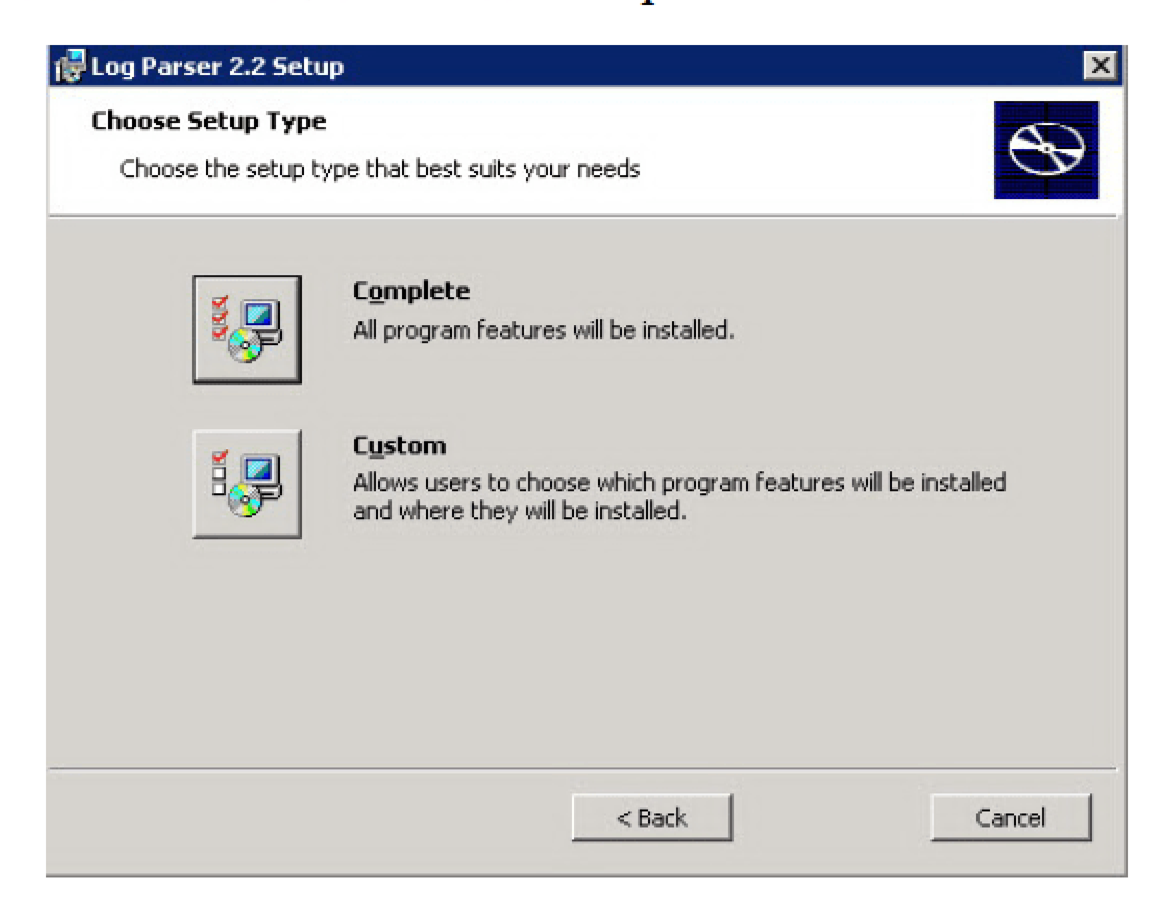
▲　安装完毕，但还要安装另外两个构件

6. 直接运行 Log Parser 的安装文件。弹出如下图所示的窗口画面，选择 Accept 之后单击 Next 按钮。



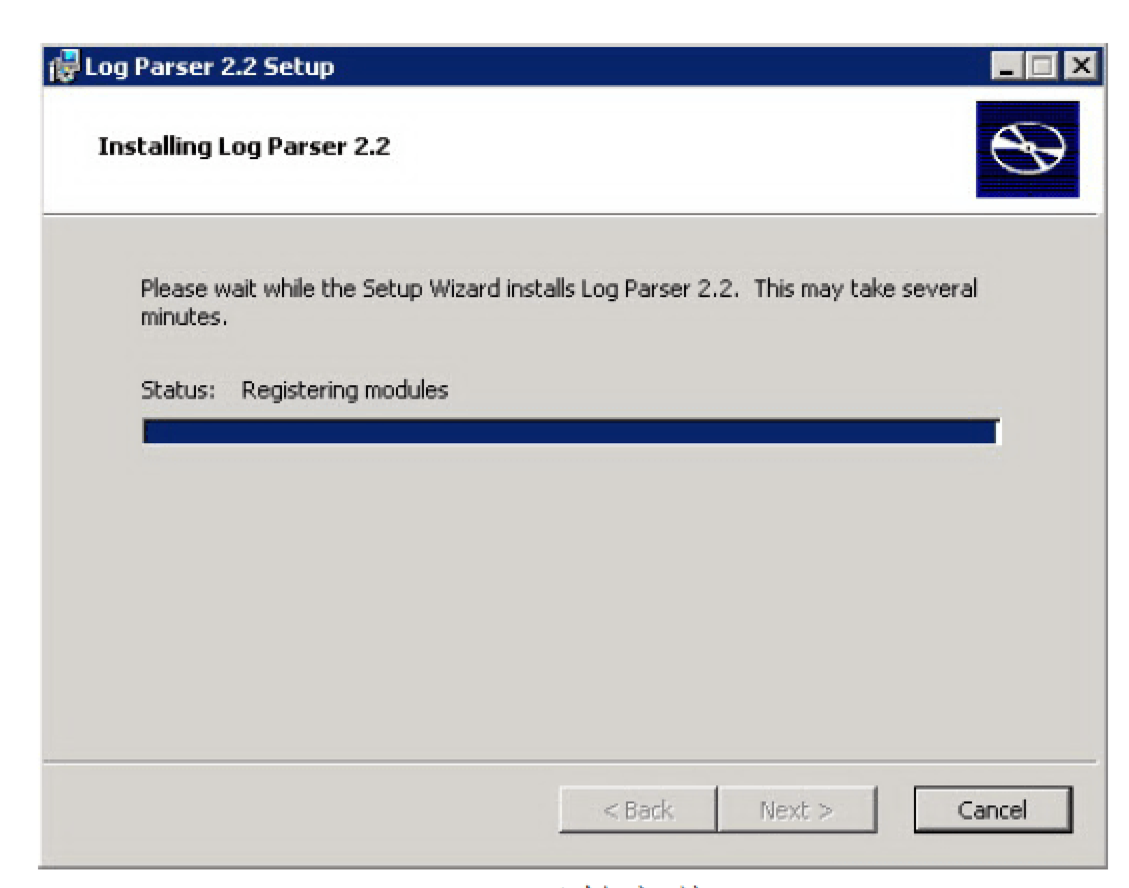
▲ 安装 Log Parser

7. 在如下图所示的窗口画面中，选择 Complete。



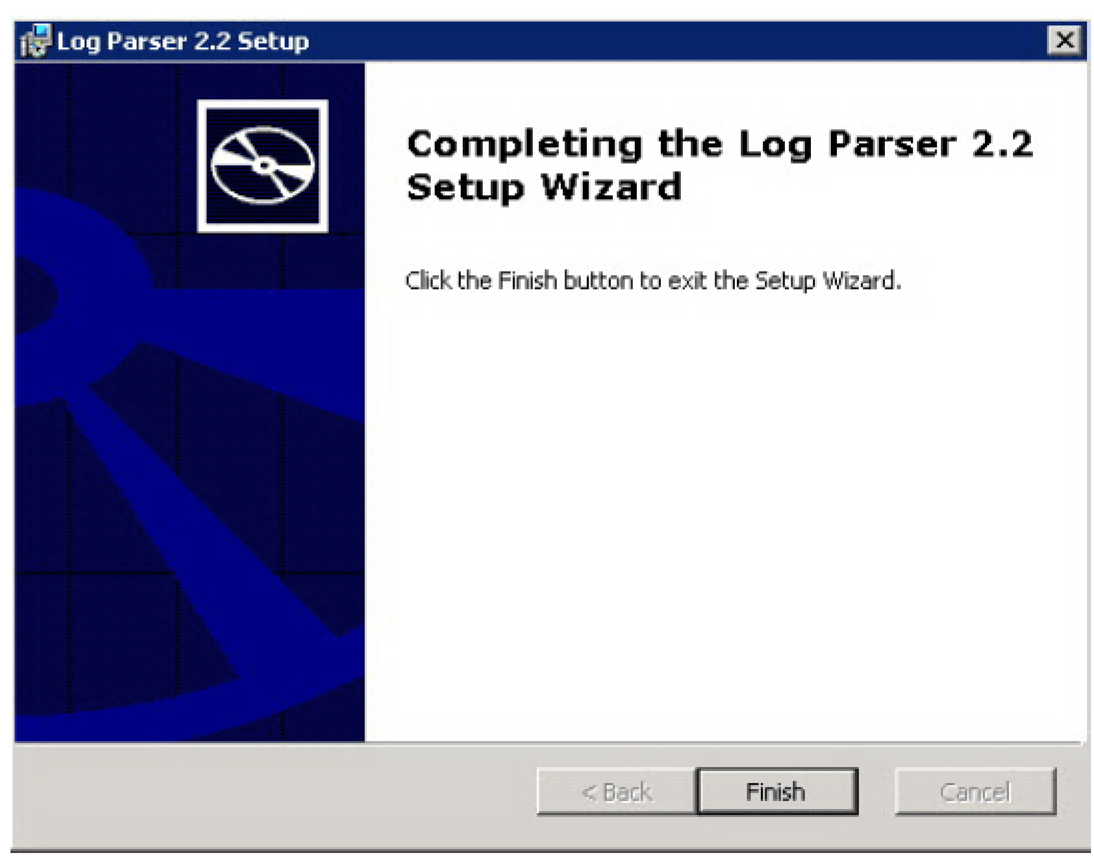
▲ 安装完整的版本

8. 接下来单击 Install 按钮，开始安装。



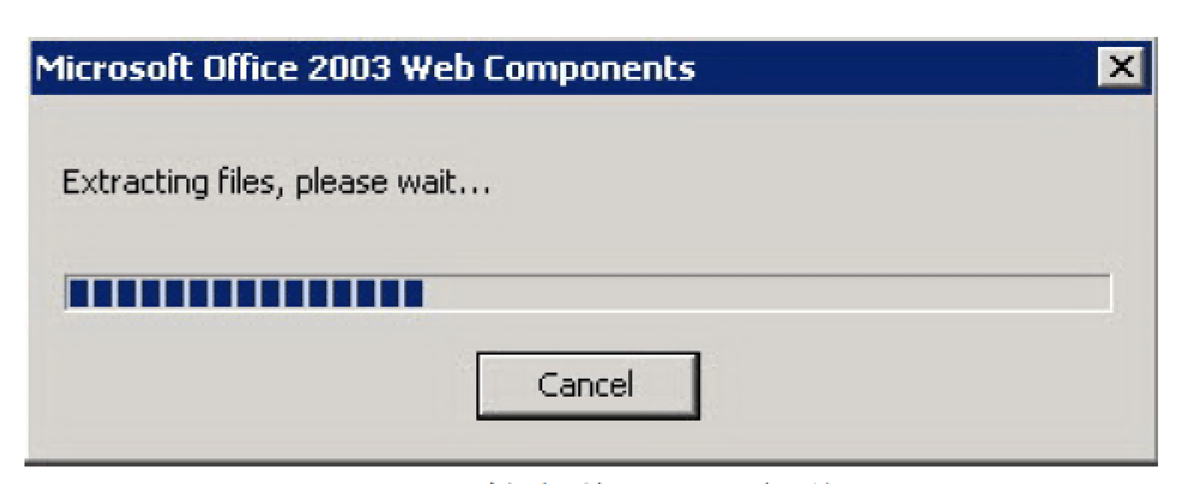
▲ 开始安装

9. 当弹出如下图所示的窗口画面时意味着安装完毕，单击 Finish 按钮即可。



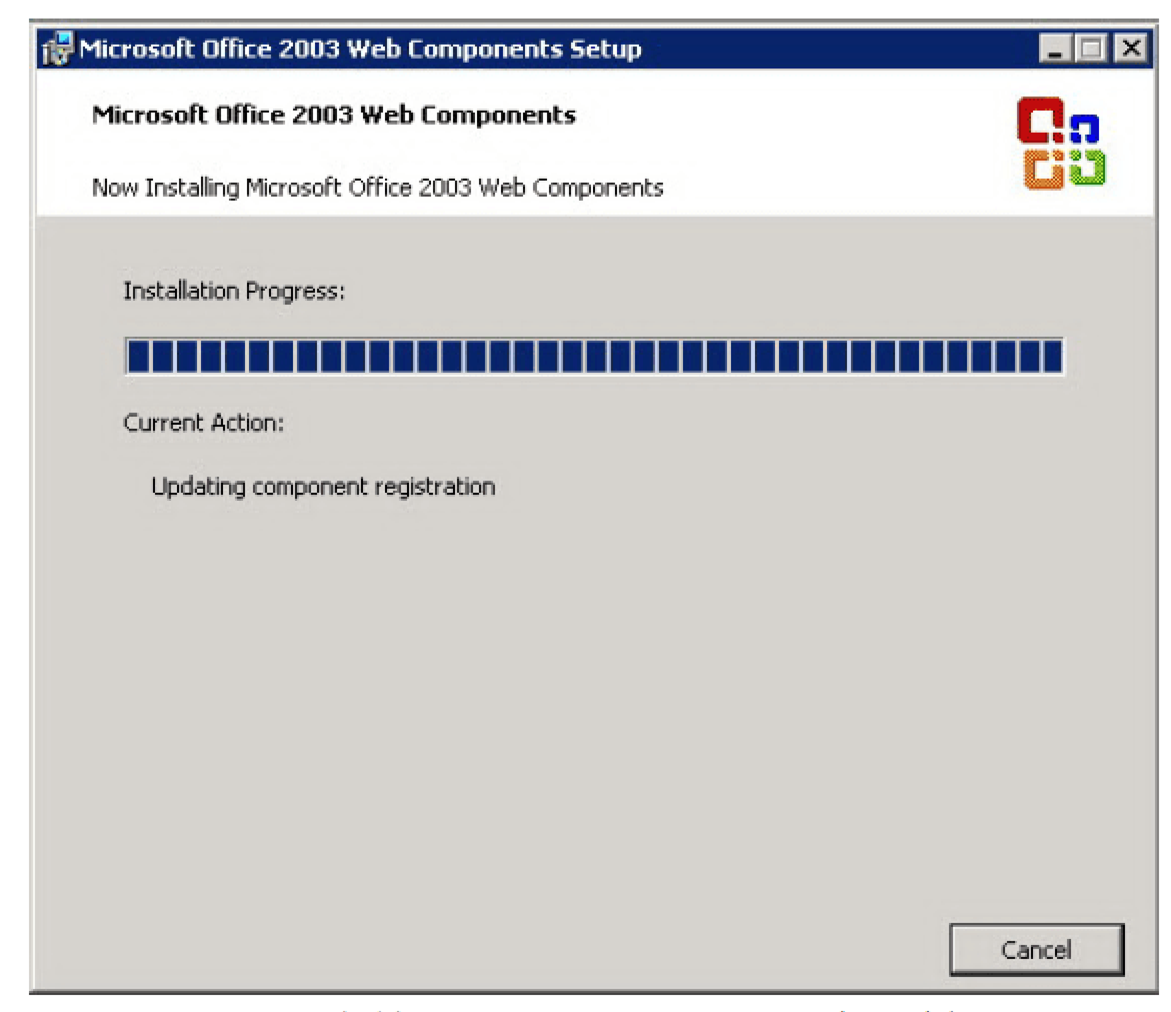
▲ 安装完毕

10. 接下来安装 Office 2003 Web Components。也是直接运行安装的文件，就会进入如下图所示的窗口的画面。



▲ 开始安装 OWC 部件

11. 选择接受使用合约，并且单击 Install 按钮。要注意的是，就算你使用的是 Office 2007，也得安装这个软件。



▲ 这是 OWC，和 Office 不太一样

12. 当弹出如下图所示的窗口的画面时，意味着安装完毕。



▲ 安装完毕了

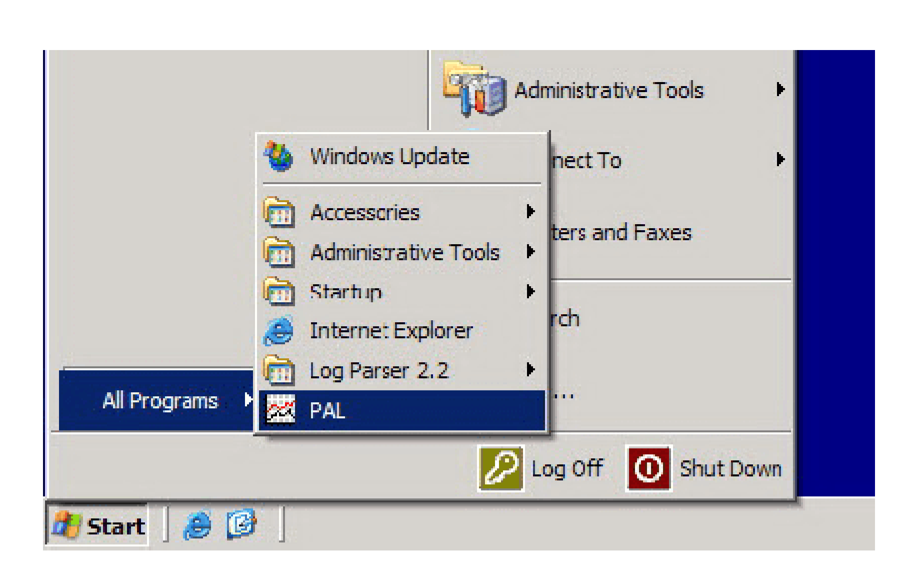
读者们接着不要忘记安装 Chat Control 及 Windows PowerShell，安装的步骤就像一般的 Windows 实用程序，直接安装即可。

4. 使用 PAL Tool

使用 PAL Tool 来提取窗口系统的 Log 文件，首先你必须有 Log 文件了，然后才可以进行提取，接下来我们就以已经表项下来的文件来进行提取，我们就来看看使用 PAL Tool 进行的提取和 Windows 本身的基本图形有何差别。

▶ 使用 PAL Tool 对系统进行效能提取

1. 首先进入 Start/PAL 运行程序。



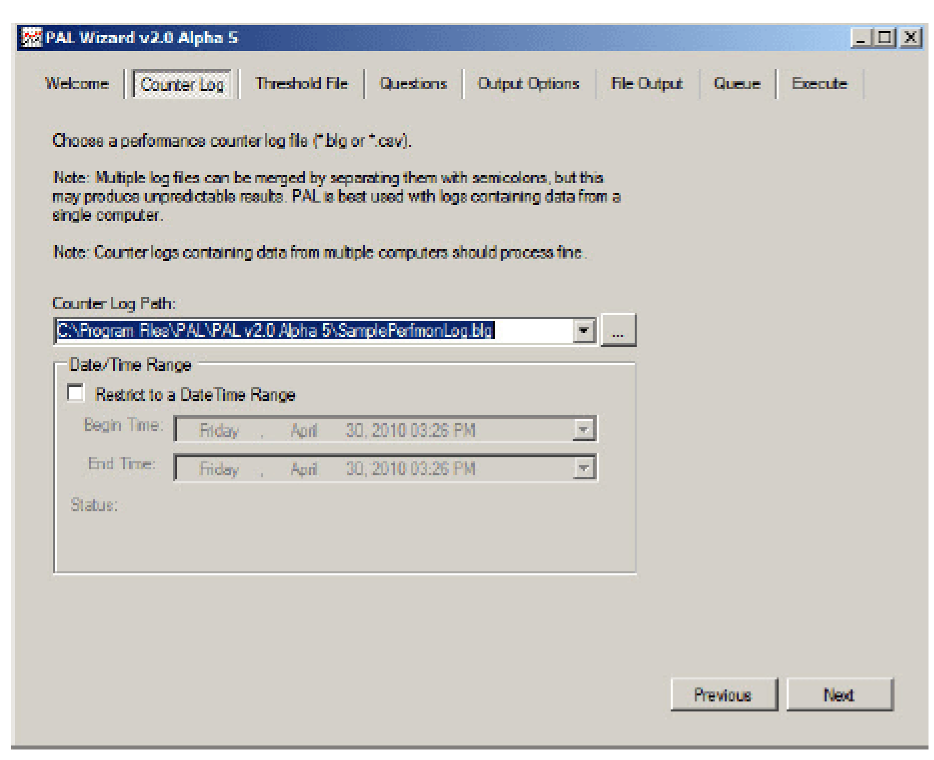
▲ 进入运行程序

2. 此时会弹出如下图所示的窗口的画面。这个窗口就是要进行 PAL Tool 的引导画面，直接单击 Next 按钮。



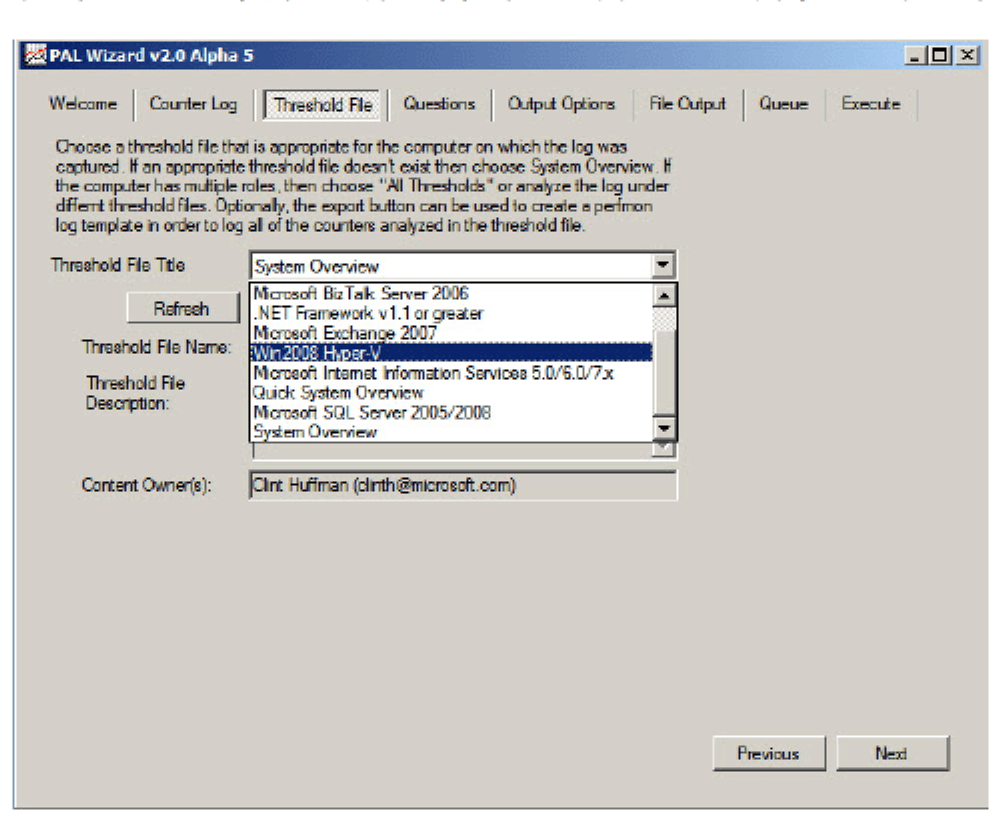
▲ 引导画面

3．下图所示为窗口系统的日志位置，一般都在 C:\PerfLog 下，在这里可以读入。当然也可以配置你要提取的时间段。选择完日志之后就按单击 Next 按钮。



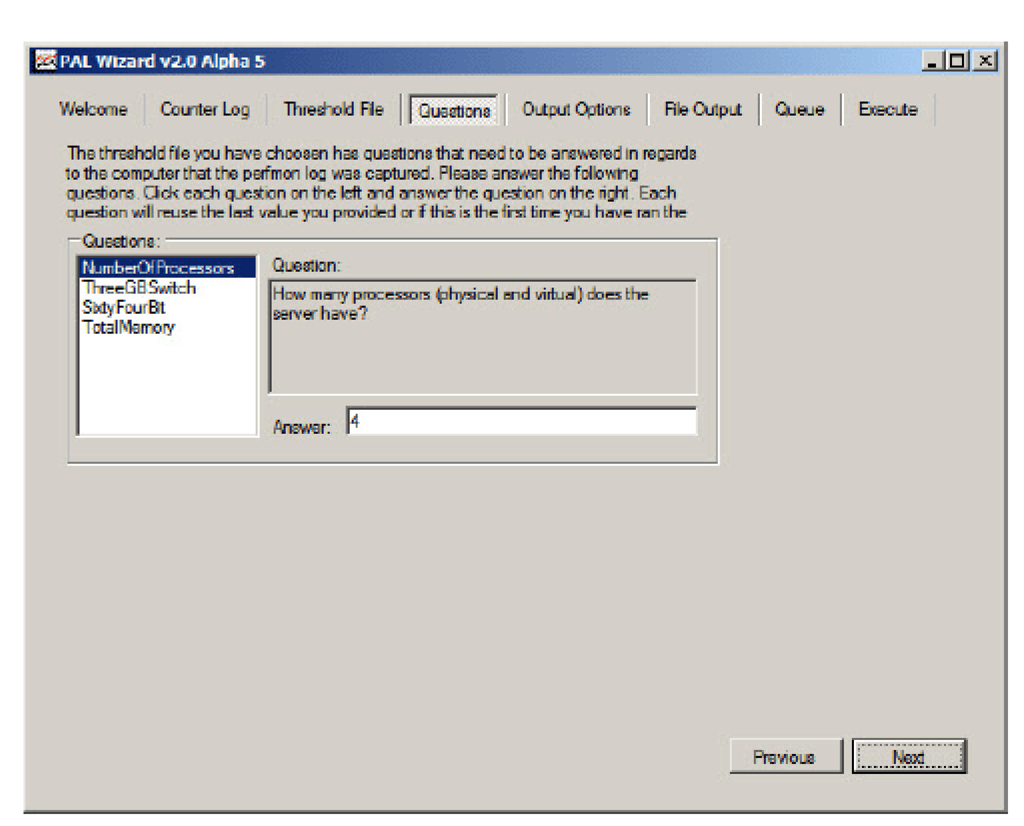
▲ 选择日志位置

4．接下来是工具中最重要的一部分。在这个窗口中，我们可以从下拉菜单中选择要评估的系统过滤项。由于 PAL Tool 中已经有不同的配置文件，可以针对不同的 Windows Server 2008 服务来配置提取的对象。举例来说，如果提取的机器是用来作为微软的 Hyper-V，就可以从这些默认的文件中选择。选择完之后单击 Next 按钮。



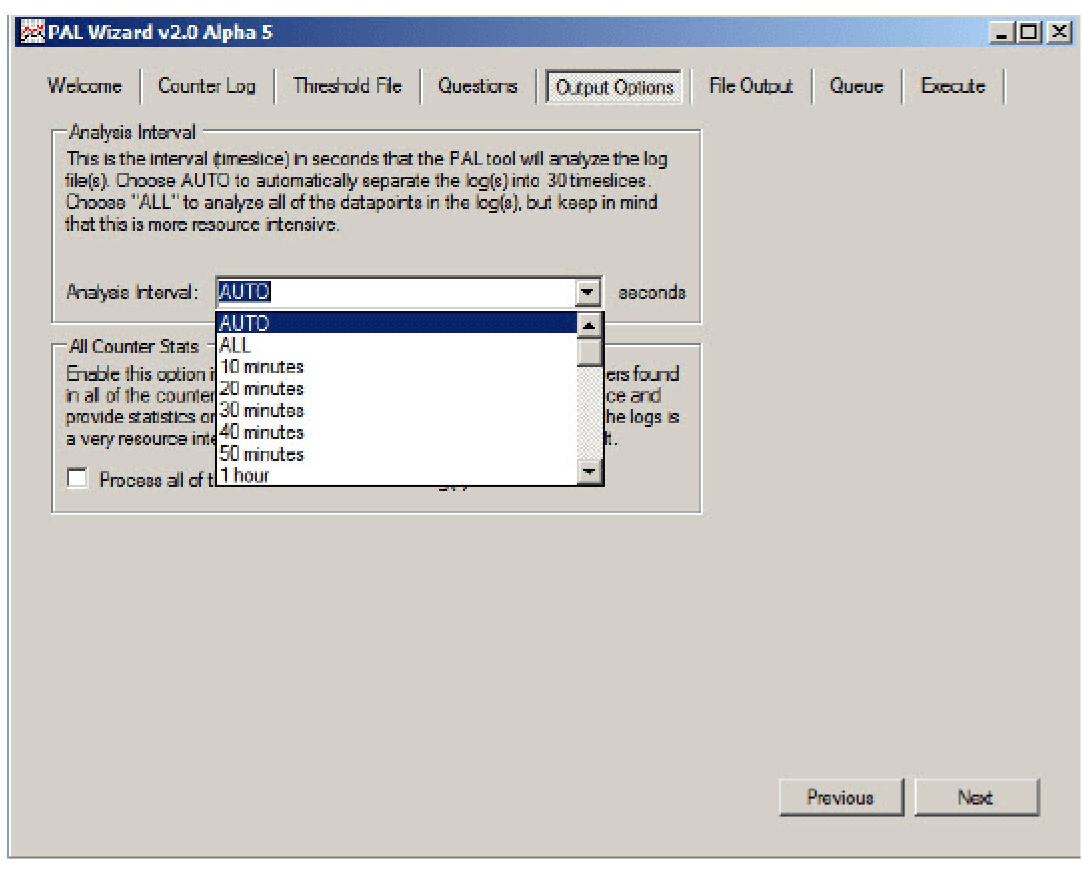
▲ 这里是选择文件

5．接下来是配置提取的选项，如使用 CPU 的数量、总内存等。之后单击 Next 按钮。



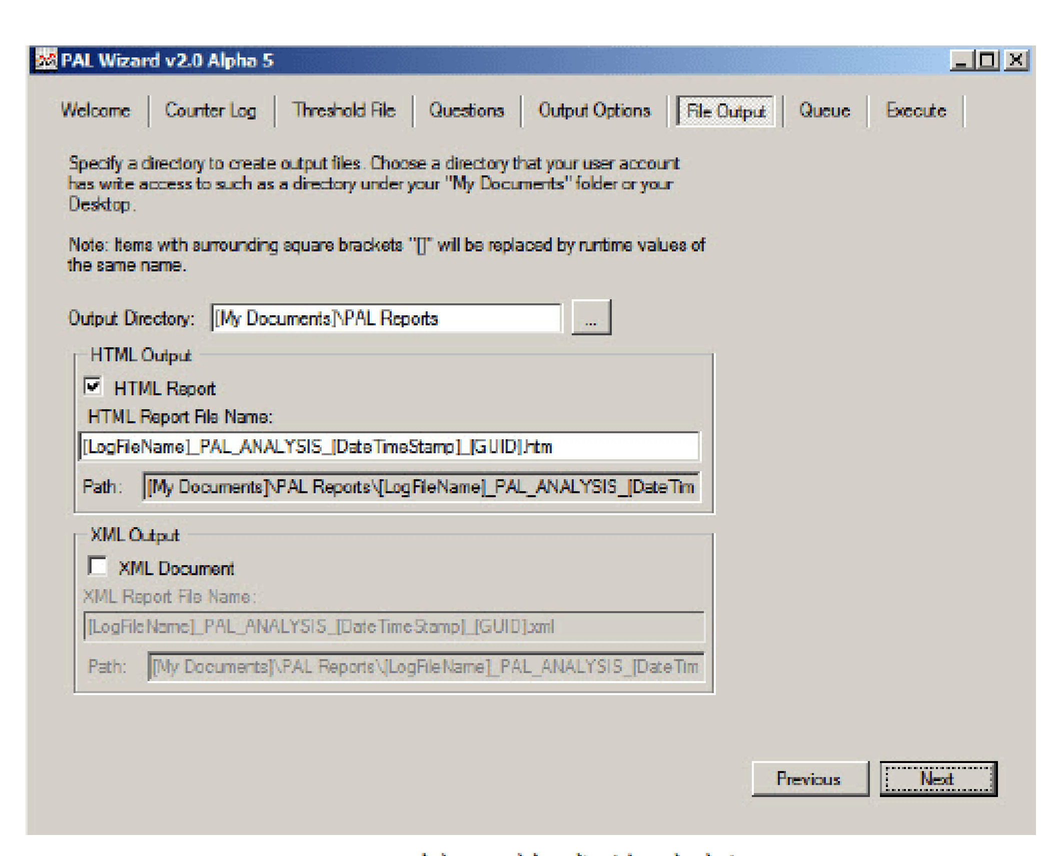
▲ 选择选项

6．接下来是输出的时间间隔，单击 Next 按钮。



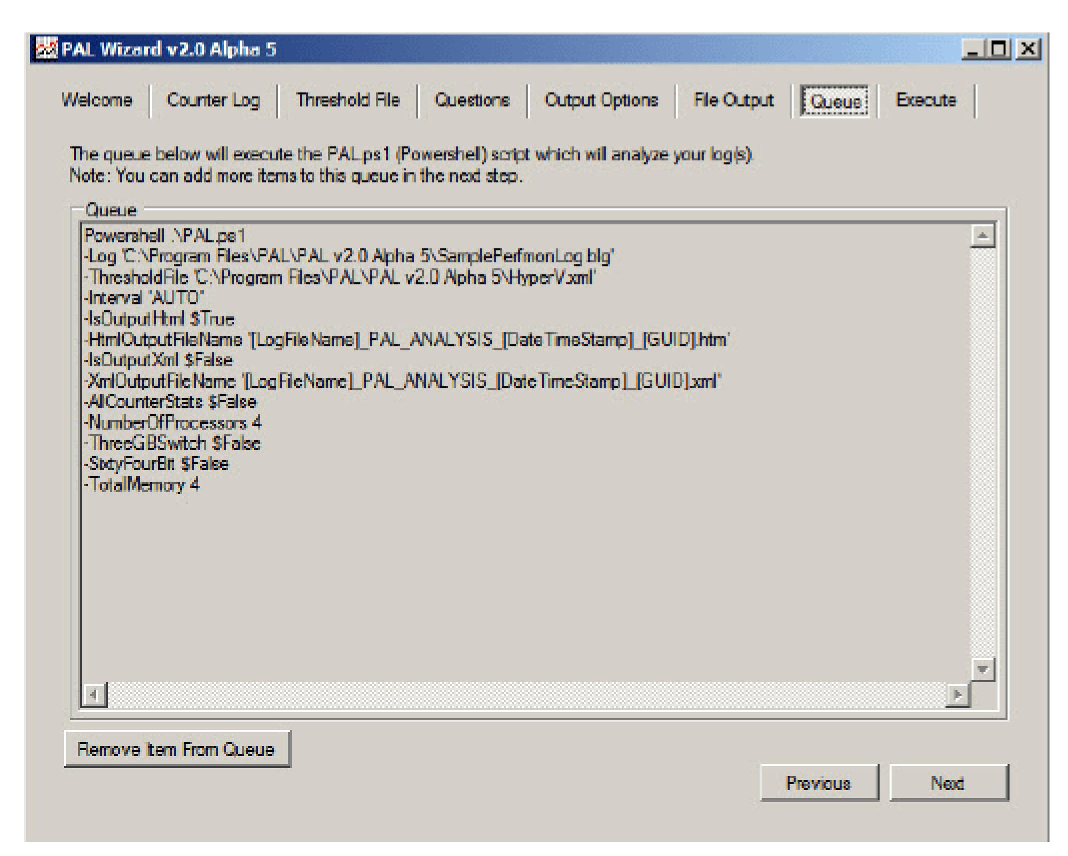
▲ 选择时间间隔

7．接下来是输出的文件档式，当然是 HTML 较方便。



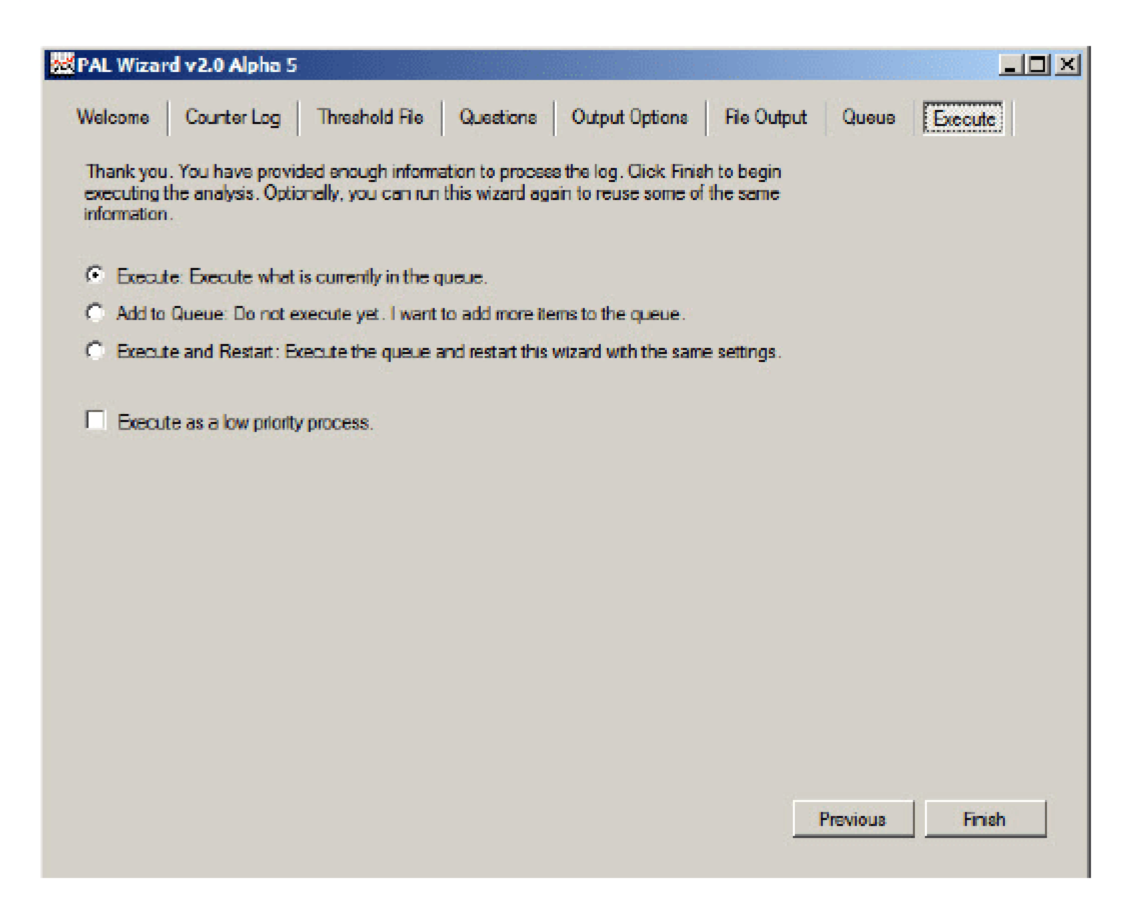
▲ 键入格式的选择

8．由于 PAL Tool 就是一个 vbscript，因此这里就是 vbscript 的参数回顾部分。如果没有要更改的地方，就直接单击 Next 按钮。



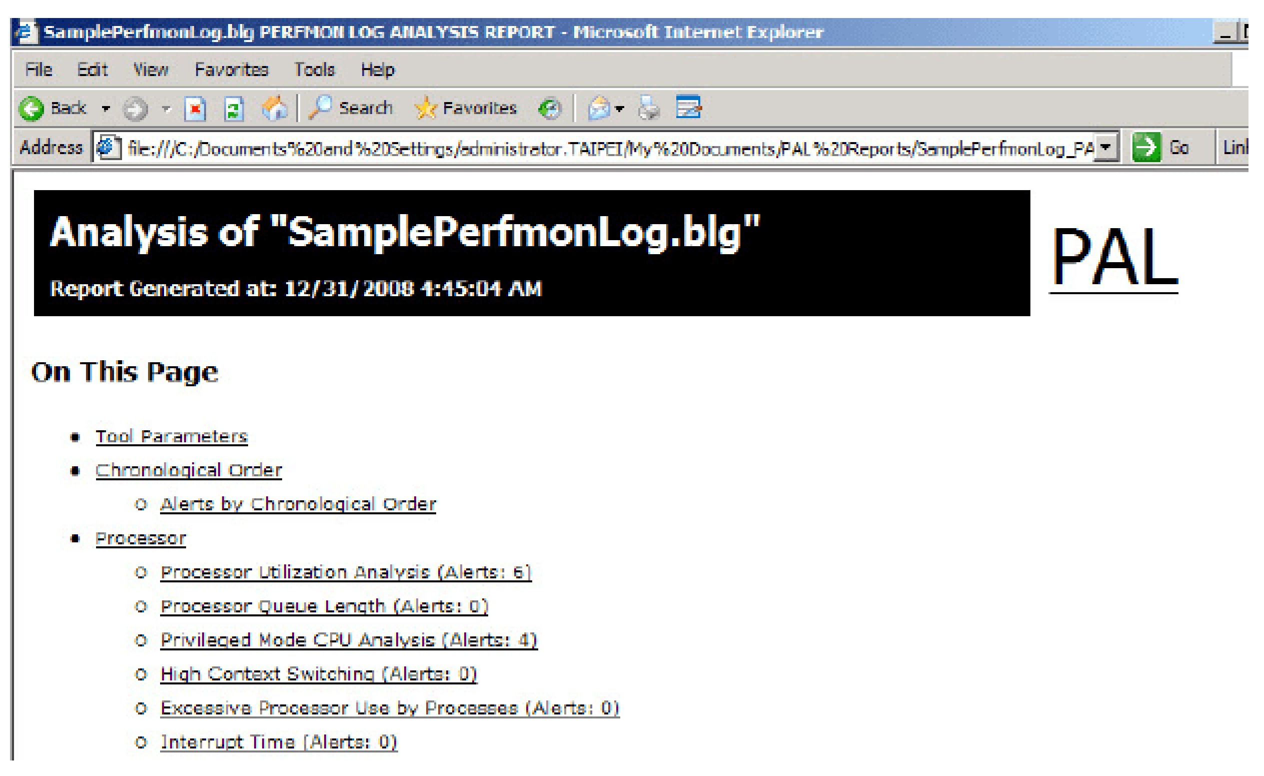
▲ 这里是输出参数

9．接下来会进入最后一个画面，就是直接运行这个 vbscript。单击 Finish 按钮就会直接开始运行提取并且输出。



▲ 如果前面的参数确定，就可以从这里运行了

10. 之后会开始进行提取，通常会花上数分钟。当弹出如下图所示的窗口时，意味着提取完毕，并且输出到给定的格式文件了。



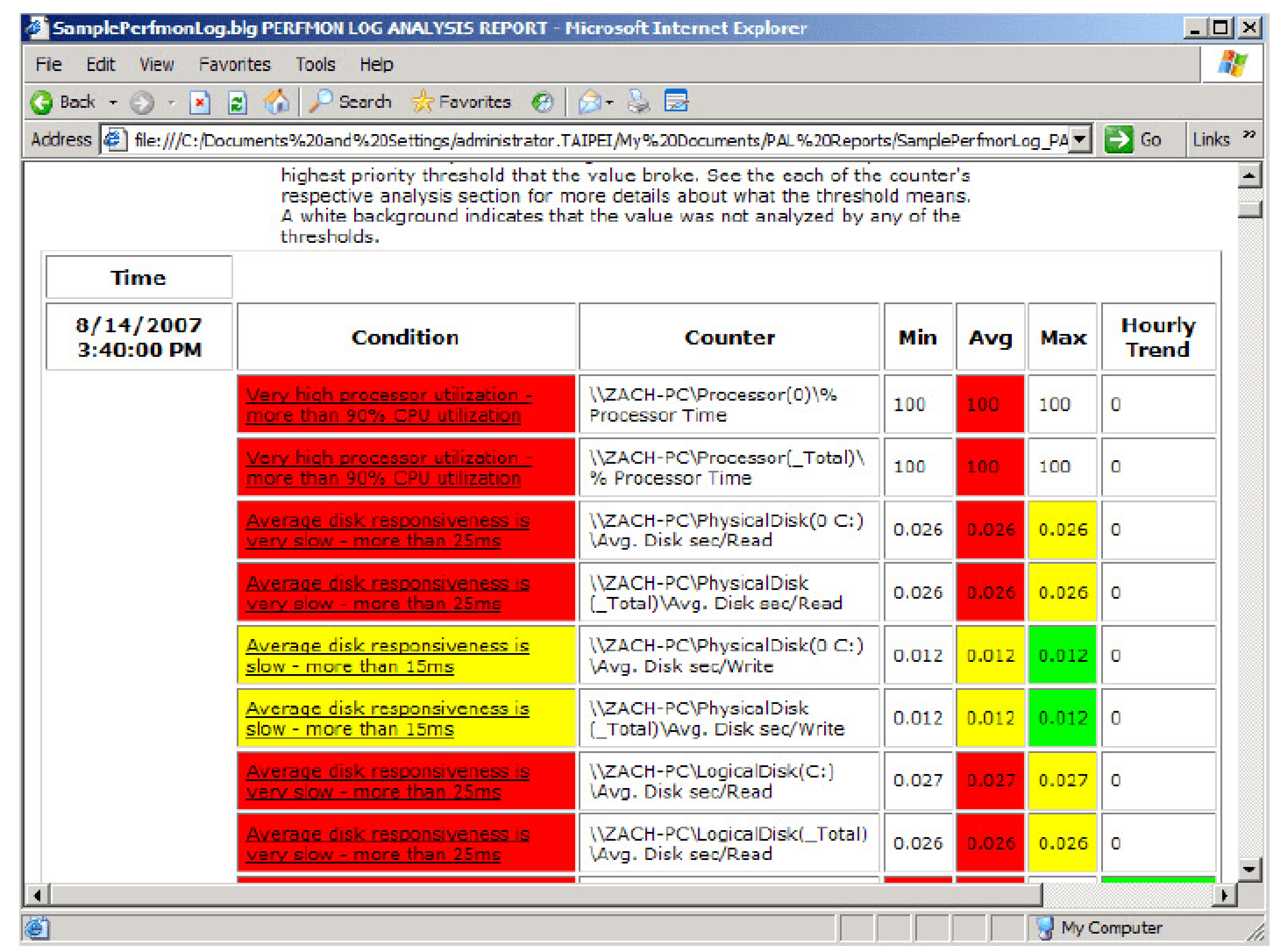
▲ 文件落实了

5. 解读 PAL Tool 输出结果

输出系统效能到了 Html 之后，就可以从窗体或颜色来看出整个系统的“瓶颈”在哪儿了。由于不同的参数配置文件有不同的内容，我们就来看看几个对虚拟机环境比较重要的参数。

▶ PAL Tool 参数说明

1. 效能警示：在整张图表中，任何红色的部分都为警示（Alerts），弹出警示时意味着该部分的效能弹出 85%以上的占有率或是弹出错误。



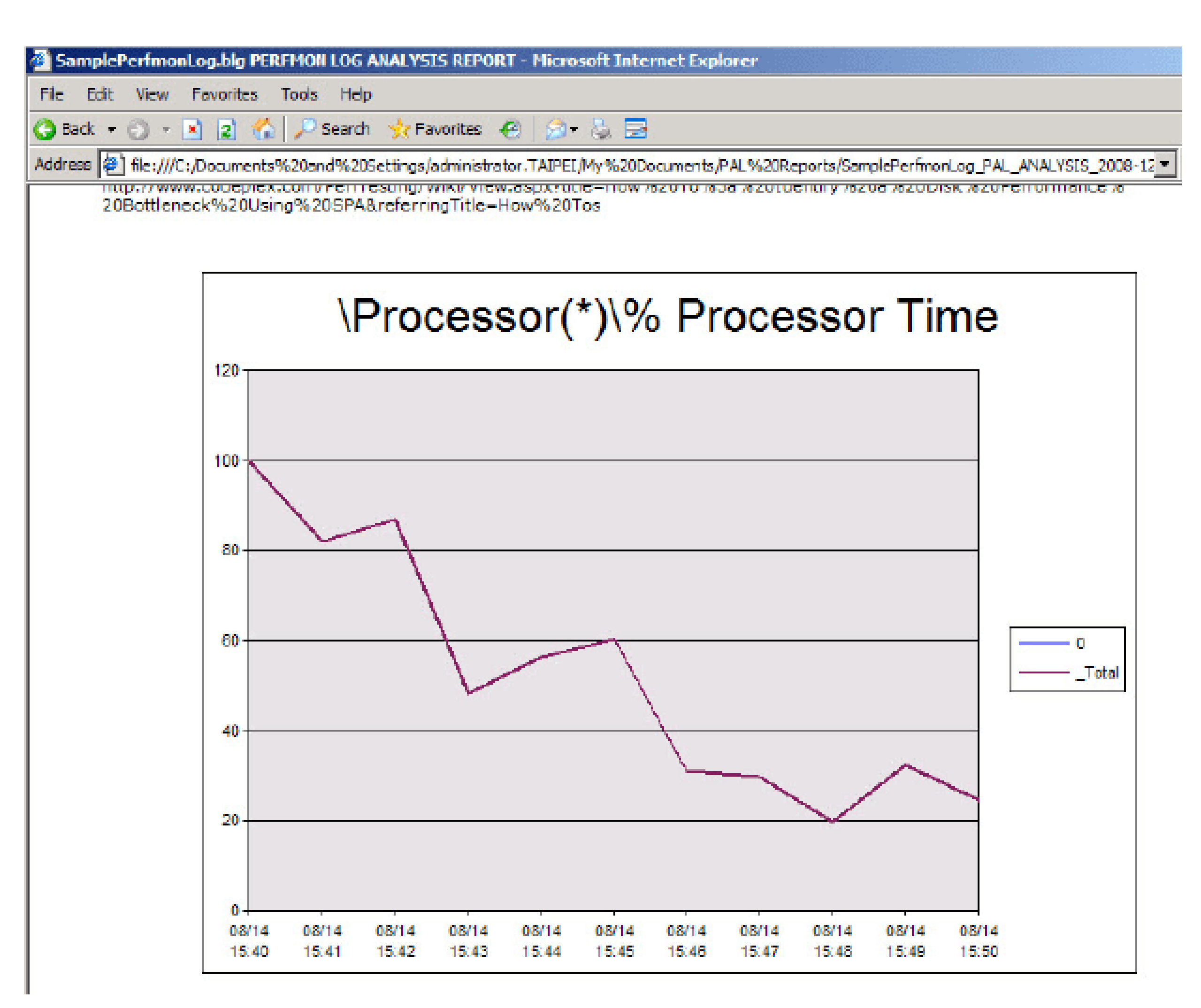
SamplePerfmonLog.blg PERFMON LOG ANALYSIS REPORT - Microsoft Internet Explorer

highest priority threshold that the value broke. See the each of the counter's respective analysis section for more details about what the threshold means. A white background indicates that the value was not analyzed by any of the thresholds.

Time	Condition	Counter	Min	Avg	Max	Hourly Trend
8/14/2007 3:40:00 PM	Very high processor utilization - more than 90% CPU utilization	\\ZACH-PC\Processor(0)\% Processor Time	100	100	100	0
	Very high processor utilization - more than 90% CPU utilization	\\ZACH-PC\Processor(_Total)\% Processor Time	100	100	100	0
	Average disk responsiveness is very slow - more than 25ms	\\ZACH-PC\PhysicalDisk(0 C:)\Avg. Disk sec/Read	0.026	0.026	0.026	0
	Average disk responsiveness is very slow - more than 25ms	\\ZACH-PC\PhysicalDisk(_Total)\Avg. Disk sec/Read	0.026	0.026	0.026	0
	Average disk responsiveness is slow - more than 15ms	\\ZACH-PC\PhysicalDisk(0 C:)\Avg. Disk sec/Write	0.012	0.012	0.012	0
	Average disk responsiveness is slow - more than 15ms	\\ZACH-PC\PhysicalDisk(_Total)\Avg. Disk sec/Write	0.012	0.012	0.012	0
	Average disk responsiveness is very slow - more than 25ms	\\ZACH-PC\LogicalDisk(C:)\Avg. Disk sec/Read	0.027	0.027	0.027	0
	Average disk responsiveness is very slow - more than 25ms	\\ZACH-PC\LogicalDisk(_Total)\Avg. Disk sec/Read	0.026	0.026	0.026	0

▲ 最基本的图示有红、黄、绿三种，报警都是红色

2. 处理器部分（Processor）：处理器最重要就是占用率。我们可以先看到一个表格，显示在这段时间内的 CPU 使用率，以虚拟机来说，长时间的平均值是最重要的，在表格之下也有文本及数字来显示 CPU 的使用率。



▲ CPU 的使用最基本的图表

Back to the top

Counter Instance Statistics

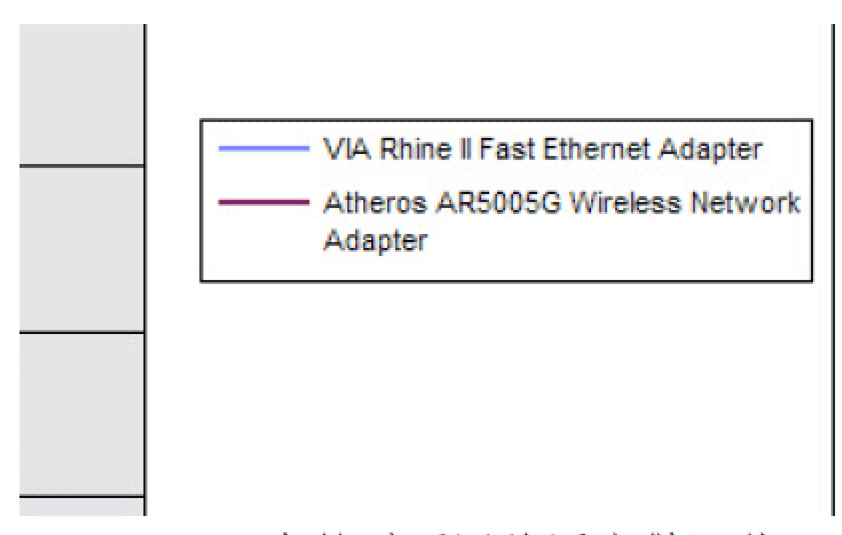
Overall statistics of each of the counter instances. Min is the minimum value recorded in the entire log, Avg is the average value of the entire log, Max is the maximum value recorded in the entire log, and Trend is the net, average, difference between data points of the entire log.

Name	Min	Avg	Max	Hourly Trend	Std Deviation	90th Percentile	80th Percentile	70th Percentile
\\ZACH-PC\Processor(0)\% Processor Time	20	52	100	-452	26	55	53	52
\\ZACH-PC\Processor(_Total)\% Processor Time	20	52	100	-452	26	55	53	52

Back to the top

▲ 也有数值的显示部分

3. 网络部分：网络部分以占用带宽为主，这对 IIS 网页服务器十分重要。当然在虚拟机中，可以将物理网卡配置给虚拟机来解决这个问题，因此网卡一般不会引起虚拟机的瓶颈问题。



▲ 会针对不同的网卡做评估

4. 磁盘部分：上一章我们也提到，要将磁盘的负担从物理服务器移到独立的存储设备，但在 HBA 的负担也会影响物理机资源的开销。在下图中可以看出磁盘系统的开销。

Counter Instance Statistics

Overall statistics of each of the counter instances. Min is the minimum value recorded in the entire log, Avg is the average value of the entire log, Max is the maximum value recorded in the entire log, and Trend is the net, average, difference between data points of the entire log.

Name	Min	Avg	Max	Hourly Trend	Std Deviation	90th Percentile	80th Percentile	70th Percentile
\\ZACH-PC\PhysicalDisk(0 C:)\Avg. Disk sec/Read	0.005	0.019	0.048	-0.122	0.014	0.02	0.018	0.017
\\ZACH-PC\PhysicalDisk(_Total)\Avg. Disk sec/Read	0.005	0.019	0.048	-0.122	0.014	0.02	0.018	0.017

Back to the top

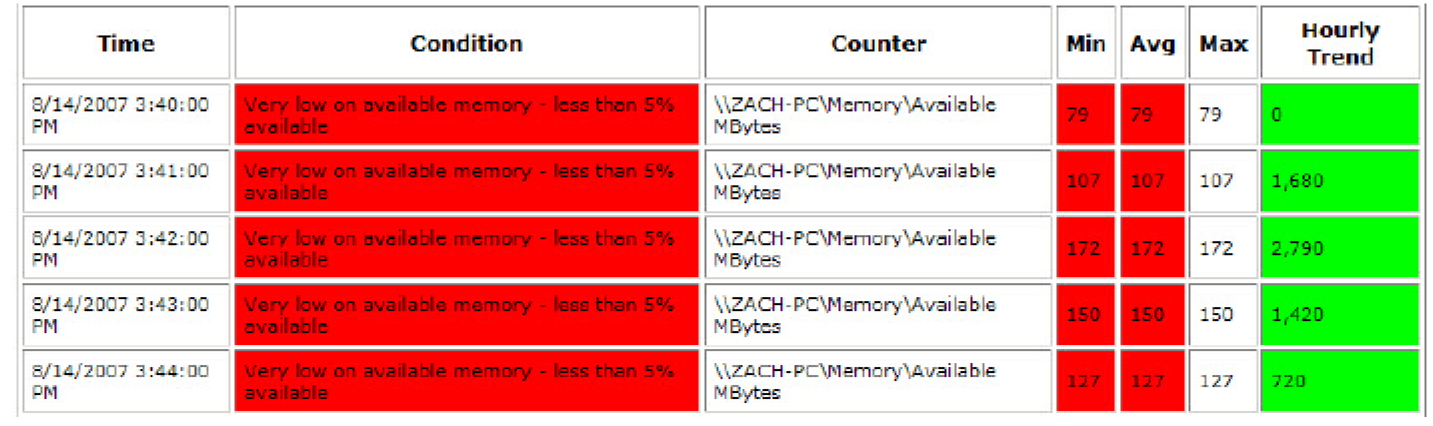
Alerts

▲ 一般磁盘是以访问时间为主的

Time	Condition
8/14/2007 3:40:00 PM	Average disk responsiveness is very slow - more than 25ms
8/14/2007 3:42:00 PM	Average disk responsiveness is slow - more than 15ms
8/14/2007 3:44:00 PM	Average disk responsiveness is very slow - more than 25ms
8/14/2007 3:45:00 PM	Average disk responsiveness is very slow - more than 25ms
8/14/2007 3:49:00 PM	Average disk responsiveness is very slow - more than 25ms
8/14/2007 3:40:00 PM	Average disk responsiveness is very slow - more than 25ms

▲ 如果超过一个访问时间就会被认为效能瓶颈，如图中的 25ms

5. 内存部分：这里的内存包括了系统内存、虚拟内存以及 Windows 最有名的 Memory Leak 问题的嗅探，当然还有内存的效率等。



Time	Condition	Counter	Min	Avg	Max	Hourly Trend
8/14/2007 3:40:00 PM	Very low on available memory - less than 5% available	\\ZACH-PC\Memory\Available MBytes	79	79	79	0
8/14/2007 3:41:00 PM	Very low on available memory - less than 5% available	\\ZACH-PC\Memory\Available MBytes	107	107	107	1,680
8/14/2007 3:42:00 PM	Very low on available memory - less than 5% available	\\ZACH-PC\Memory\Available MBytes	172	172	172	2,790
8/14/2007 3:43:00 PM	Very low on available memory - less than 5% available	\\ZACH-PC\Memory\Available MBytes	150	150	150	1,420
8/14/2007 3:44:00 PM	Very low on available memory - less than 5% available	\\ZACH-PC\Memory\Available MBytes	127	127	127	720

▲ 内存在虚拟机中是一对一的，所以物理机多少内存，虚拟机就多少，但虚拟机中是可以压缩内存的，因此还是要知道内存的开销量

6.2 Linux 系统效能的评估

在 Linux 下要监测服务器各项性能指针有太多选择了，最常听到就是 MRTG。本节的目的不是深入研究 MRTG，因为大家都知道，在 Linux 下的任何一个应用的功能都强大得不知怎么办，要理解每一个功能是不太可能的，因此我们只要理解到和虚拟机相关的部分就行。

```
Processes: 86 total, 2 running, 84 sleeping, 410 threads
Load Avg: 0.28, 0.39, 0.40  CPU usage: 8.83% user, 8.37% sys, 82.79% idle  SharedLibs: 8260K resident, 8996K data, 0B linkec
PhysMem: 690M wired, 1091M active, 391M inactive, 2971M used, 1125M free. VM: 163G vsize, 1041M framework vsize, 116256(0) p
Disks: 81351/1742M read, 62325/1028M written.

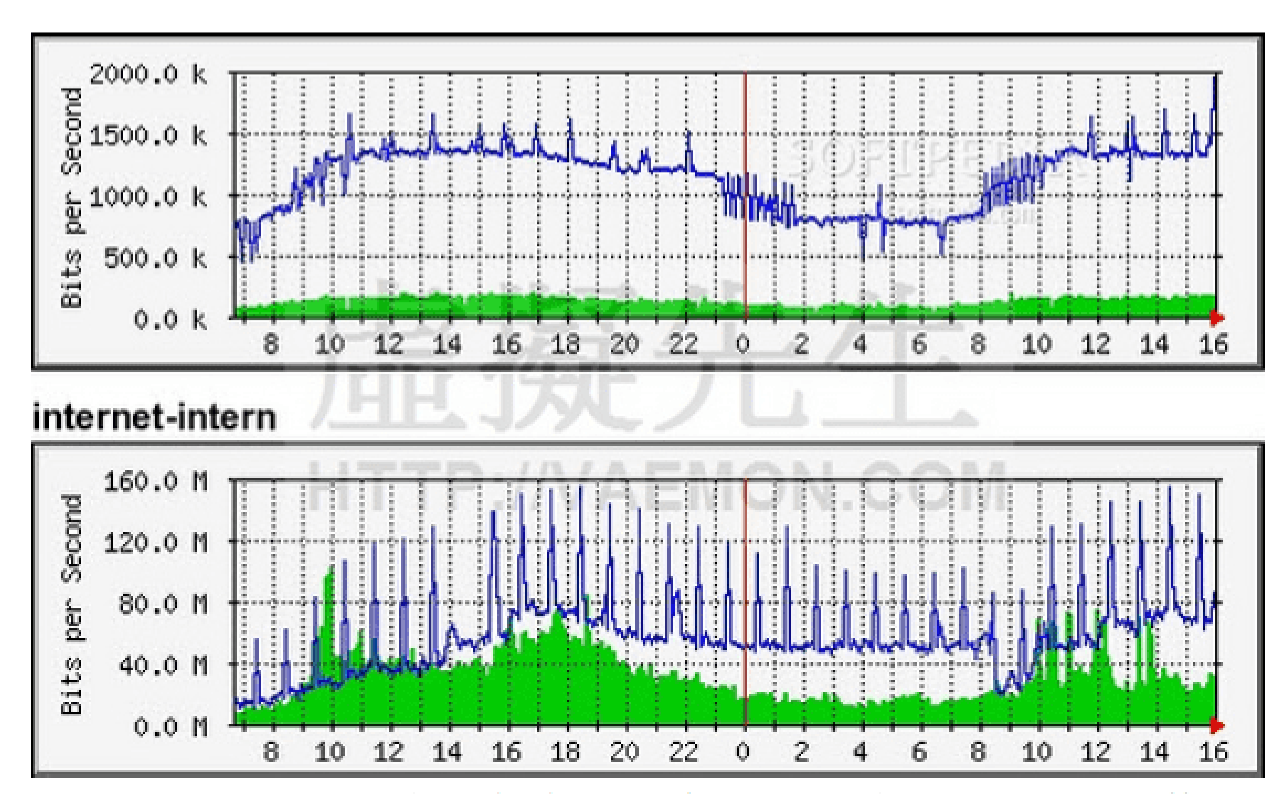
PID   COMMAND          %CPU TIME     #TH  #WQ  #POR #MREG RPRVT  RSHRD  RSIZE  VPRVT  VSIZE  PGRP PPID STATE    UID FAULTS
1427  screencapture    1.6  00:00.05 3    2    42   79    492K+  19M-   2956K  14M-   2666M  743  1    sleeping 501 1248+
1425  top              6.3  00:01.96 1/1  0    25   33    2304K  264K   2880K  17M    2378M  1425 1134 running  0   19404+
1422  mdworker         0.0  00:00.05 3    1    47   56    1336K  19M    3816K  21M    2403M  1422 1    sleeping 89  1585
1169- MAMP             0.0  00:00.44 2    1    92   117   5012K  30M    9244K  31M    905M   1169 684  sleeping 501 4279
1148- Adobe Updater    0.0  00:00.19 2    1    88   210   1932K  30M    6176K  30M    939M   1148 684  sleeping 501 2549
1143- Adobe Photoshop  1.1  03:38.82 10   1    241  1288  81M    84M    129M   151M   1274M  1143 684  sleeping 501 190262
1134  bash             0.0  00:00.03 1    0    17   24    416K   244K   1084K  17M    2378M  1134 1133 sleeping 501 641
1133  login            0.0  00:00.01 1    0    22   53    476K   244K   1584K  18M    2379M  1133 1122 sleeping 0   619
1132  ssh-agent        0.0  00:00.06 2    1    33   60    1312K  672K   2772K  39M    2400M  1132 684  sleeping 501 1056
1122  Terminal         2.0  00:15.11 5    1    113  116   5664K  33M    17M    34M    2711M  1122 684  sleeping 501 28699+
1080  Preview          0.0  00:23.93 2    1    105  187   8192K  41M    32M    27M    2731M  1080 684  sleeping 501 34587
1058  java             0.0  00:19.80 29   1    244  253   98M    8016K  131M   264M   2735M  1052 1052 sleeping 501 65961
1052  eclipse          0.9  01:51.85 32   2    389  605   262M   48M    317M   458M   3702M  1052 684  sleeping 501 220360
1042  AppleSpell       0.0  00:00.05 2    1    34-  41-   708K-  10M    1820K- 28M-   2402M- 1042 684  sleeping 501 951
1033- Wallet           0.0  00:12.57 4    2    767  337   6956K  52M    22M    33M    967M   1033 684  sleeping 501 80837
759-  firefox-bin      5.8  44:00.65 19   1    950  6658  419M   77M    578M   469M   1726M  759  684  sleeping 501 3430287
756   mdworker         0.0  00:06.19 3    1    50   137   9276K  17M    21M    43M    2430M  756  1    sleeping 501 43916
753-  VMware Fusion St 0.0  00:00.23 3    1    89   85    1104K  19M    4080K  30M    893M   753  684  sleeping 501 2278
747*  MagicMenuHotKeyD 0.0  00:01.13 4    0    86   154   6800K  10M    11M    96M    961M   747  721  sleeping 501 4694
743   SystemUIServer   0.0  00:02.20 3    1    282  238   11M    26M    17M    37M    2700M  743  684  sleeping 501 16117
739   VDCAssistant     0.0  00:00.18 4    1    98   90    1616K  22M    5504K  32M    2679M  739  684  sleeping 501 2994
731   iChatAgent       0.0  00:02.69 5    1    80   219   3144K  15M    10M    16M    2669M  731  684  sleeping 501 6807
730-  GrowlMenu        0.0  00:00.26 2    1    71   164   1680K  17M    4460K  30M    920M   730  684  sleeping 501 3012
729-  VMware Fusion He 0.0  00:27.87 6    2    626  180   7036K  23M    18M    38M    929M   729  684  sleeping 501 9317
728-  Snapz Pro X      0.2  00:54.18 2    1    93   310   5340K  45M    13M    36M    973M   728  684  sleeping 501 9818
```

▲ 一般在 Linux 下是使用 top 来检视性能，但无法进行长时间表项

6.2.1 MRTG 简介

MRTG 全称为多重路由流量表（Multi Router Traffic Graphic），利用图表的方式显示主机上一些指针的效能，可以通过 Apache 或 Tomcat，用网页的图表方式来检视。虽然 MRTG 在一开始是用来检视网络流量的，但在使用不同的定义文件来生成数据之后，也可以显示其他数据在 MRTG 的图表上，如 CPU、RAM、DISK 等方面。

MRTG 原来的用处就是看网络的流量，因此在 Linux 下，会以默认的网卡（eth0）来作为显示流量，也就是说，本小节所有的步骤都落实后，MRTG 就直接可以检视 Linux 下第一片网卡 eth0 的流量了。



▲ MRTG 能做的事情非常多，如网络流量及 CPU 等

▶ 安装 MRTG 的准备任务

在安装 MRTG 之前，你必须作好下面的准备。

（1）对 Linux 有基本的认识，如安装 binary 套件、权限、文件等观念。

（2）会使用 vi 编辑器。

（3）Linux 已安装 LAMP 环境。

（4）Linux 的网卡可常规使用，取得 IP 可和外网连接。

（5）能从网上或光盘机补丁套件。

6.2.2 开始安装 MRTG

当你准备好使用 Linux 之后，就可以安装 MRTG 了，MRTG 是由下列几个套件组成的，因此必须配置下列的套件。

▶ 安装 MRTG 的步骤列表

（1）安装 snmpd 服务以及 snmp 工具条程序。

（2）配置 snmpd 参数。

（3）测试 snmpd 参数。

（4）安装 mrtg 套件。

（5）配置 mrtg 的运行环境。

（6）创建 mrtg 的配置文件。

（7）以配置文件中的参数来运行 mrtg。

（8）创建 Apache 下的 mrtg 目录。

（9）创建 Apache 下的 index.html 文件。

（10）从网页检视 mrtg 的图表。

1. 安装 snmpd 服务以及 snmp 工具条程序

snmp 是标准的网络监控协议，在 Linux 和 Windows 下都有支持。在安装了 snmp 协议之后，就可以监视主机的数据了。Linux 下使用的 snmpd 这个服务，但在安装 Ubuntu 时是没有安装的，因此我们要手动安装。

▶ 安装 snmpd 服务及工具

1. 进入 Linux 的命令提示列下，并且键入 sudo apt-get install snmpd snmp。
2. 键入用户口令之会弹出安装的画面，安装时系统会要求键入 y，意味着同意安装，之后就会进行一些说明，在安装完毕时回到命令行提示下。

```
vaemon@vaemon:/$
vaemon@vaemon:/$ sudo apt-get install snmpd snmp
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  libperl5.10 libsensors3 libsnmp-base libsnmp15 libsysfs2
Suggested packages:
  lm-sensors
The following NEW packages will be installed:
  libperl5.10 libsensors3 libsnmp-base libsnmp15 libsysfs2 snmp snmpd
0 upgraded, 7 newly installed, 0 to remove and 11 not upgraded.
Need to get 2,239kB of archives.
After this operation, ,590kB of additional disk space will be used.
Do you want to continue [Y/n]? _
```


▲ 在 Linux 下安装 snmpd 及 snmp 的套件

3. 当安装完毕时，snmpd 就会激活。我们可以用 ps aux | grep snmpd 命令检视是否有激活。

```
vaemon@vaemon:/$
vaemon@vaemon:/$ ps aux  grep snmpd
snmp      7676  0.0  0.9                ?       S    11:47   0:00 /usr/sbin/snmp
-Lsd -Lf /dev/null -u snmp -I -smux -p /var/run/snmpd.pid 127.0.0.1
vaemon    7685  0.0  0.1   7336   876 tty1      R+   11:51
to snmpd
vaemon@vaemon:/$ _
```

▲ 检视 snmpd 是否激活

在运行服务中有 snmpd 这个服务，意味着 Linux 主机上的 snmpd 安装并且成功激活了。

2. 配置 snmpd 参数

接下来要配置 snmpd 参数，以求 snmpd 在 Linux 下能正常运行。在 Ubuntu 下，snmpd 的参数配置文件放在/etc/snmp/snmpd.conf 下，我们就直接编辑这个文件即可。

▶ 配置 snmpd 参数

1. 键入 sudo vim /etc/snmp/snmpd.conf。

```
#
# For more information, read the FAQ as well as the snm
# manual page.

####
# First, map the community name (COMMUNITY) into a secu
# (local and mynetwork, depending on where the request
# from):

#        sec.name  source          community
com2sec paranoid  default         public
#com2sec readonly  default         public
#com2sec readwrite default         private

####
# Second, map the security names into group names:
```

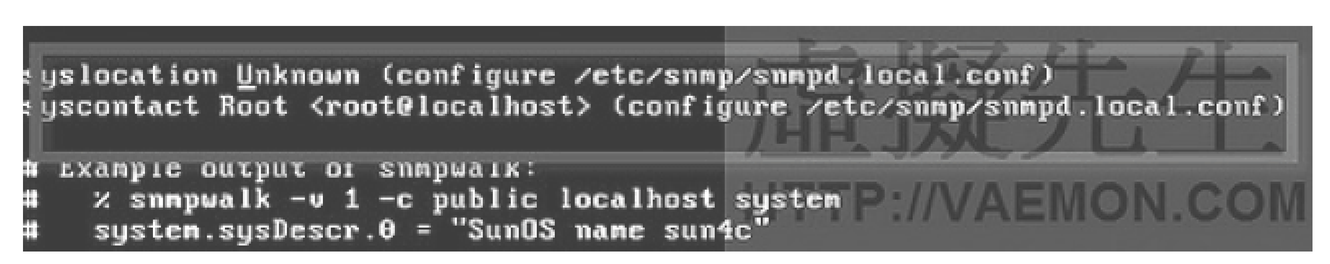
▲ 先找到此画面

2. 进入之后，找到下图所示的地方，改成我们要的样子。

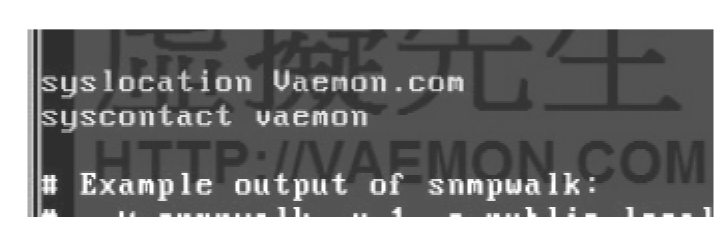


▲ 将#号拿掉

3. 之后找到下一个图的地方，这里是一些辨识用的数据，也改成如图中的例子。



▲ 这是辩视数据，在 MRTG 的网页上会显示



▲ 改成你要的样子

更改完之后，存盘离开 vi，并且重新激活 snmpd 服务，还是键入 sudo /etc/init.d/snmpd restart。弹出如下图所示的画面就意味着重新激活成功了。



▲ 重新激活 snmpd 即可

3. 测试 snmp 参数

刚才安装了 snmp 套件，就是要用其中的 snmpwalk 的命令来测试 snmpd 是否成功。我们键入 snmpwalk –v 1 –c public localhost system，如果弹出了一大堆参数，意味着 snmpd 配置已经成功了。如果弹出 timeout 或是 no response 之类的消息，意味着 snmpd 没有安装好。这时要遍历网卡是否有激活，遍历 snmpd 是否在服务中激活等。一般来说，snmpd 不太会出问题，所以如果这个参数有问题，要多看看其他网络、IP、服务等问题。

```
vaemon@vaemon:/$ snmpwalk -v 1 -c public localhost system
SNMPv2-MIB::sysObjectID.0 = OID: NET-SNMP-MIB::netSnmpAgentOIDs.10
DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (1332) 0:00:13.32
SNMPv2-MIB::sysContact.0 = STRING: vaemon
SNMPv2-MIB::sysName.0 = STRING: vaemon
SNMPv2-MIB::sysLocation.0 = STRING: Vaemon.com
SNMPv2-MIB::sysORLastChange.0 = Timeticks: (0) 0:00:00.00
SNMPv2-MIB::sysORID.1 = OID: SNMP-FRAMEWORK-MIB::snmpFrameworkMIBCompliance
SNMPv2-MIB::sysORID.2  public localhost system
#    system.sysDescr.0 = "SunOS name sun4c"
#    system.sysObjectID.0 = OID: enterprises.ucdavis.ucdSnmpAgent.sunos4
#    system.sysUpTime.0 = Timeticks: (595637548) 68 days, 22:32:55
#    system.sysContact.0 = "Me <me@somewhere.org>"
#    system.sysName.0 = "name"
"/etc/snmp/snmpd.conf" 414L, 15846C written
vaemon@vaemon:/$ sudo /etc/init.d/snmpd restart
 * Restarting network management services:                          [ OK ]
vaemon@vaemon:/$ snmpwalk -v 1 -c public localhost system
SNMPv2-MIB::sysDescr.0 = STRING: Linux vaemon 2.6.31-14-server #48-Ubuntu SMP Fr
i Oct 16 15:07:34 UTC 2009 x86_64
```

▲ 弹出图中的消息即可

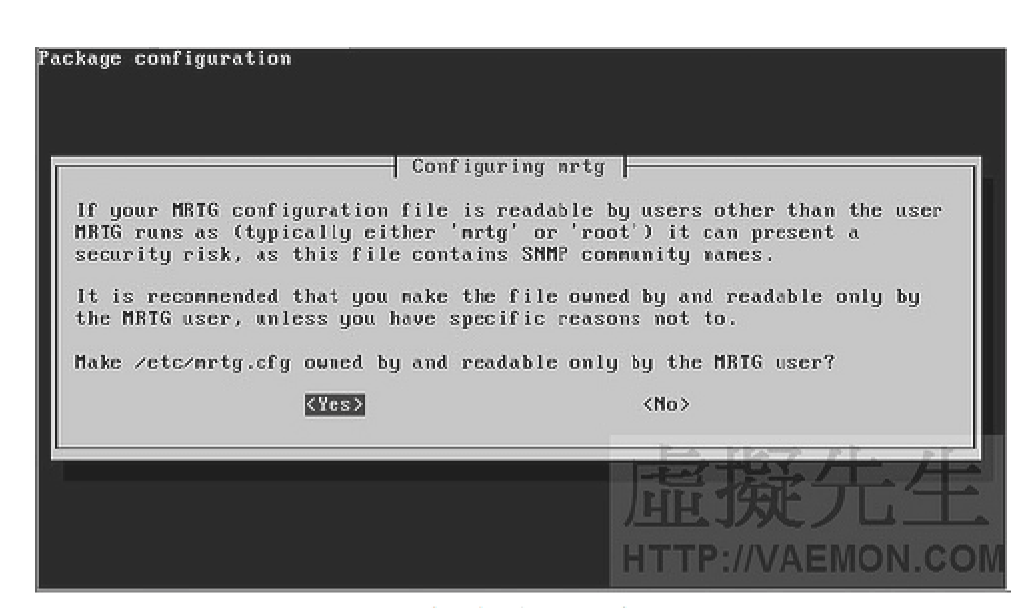
4. 安装 mrtg 套件

安装 mrtg 就像安装任何 Ubuntu 下的套件一样，用 apt-get 就行。只要键入 sudo apt-get install mrtg。

```
vaemon@vaemon:/$
vaemon@vaemon:/$
vaemon@vaemon:/$ sudo apt-get install mrtg
[sudo] password for vaemon:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  libgd2-noxpm libio-socket-inet6-perl libjpeg62 libpng12-0
  libsnmp-session-perl libsocket6-perl
Suggested packages:
  libgd-tools mrtg-contrib
The following NEW packages will be installed:
  libgd2-noxpm libio-socket-inet6-perl libjpeg62 libpng12-0
  libsnmp-session-perl libsocket6-perl mrtg
0 upgraded, 7 newly installed, 0 to remove and 11 not upgraded.
Need to get 1,041kB of archives.
After this operation, 3 027kB of additional disk space will be used.
Do you want to continue [Y/n]? _
```

▲ 键入安装，会要求安装套件，键入 y 即可

安装时，系统会询问是否要用 mrtg 的用户方式来运行。只要单击 Yes 按钮即可。这么做的目的就是为了系统安全。在安装完毕之后，系统会在 Linux 下创建一个 mrtg 的 cronjob，每 5 分钟会运行一次 mrtg 任务来补丁监视的条目，默认值就是 eth0 网卡的流量。这个步骤落实后，意味着 mrtg 监视安装完毕了，但 mrtg 的监视结果还不知道，我们必须创建能从 Apache 上观察的 mrtg 网页和配置文件才行。



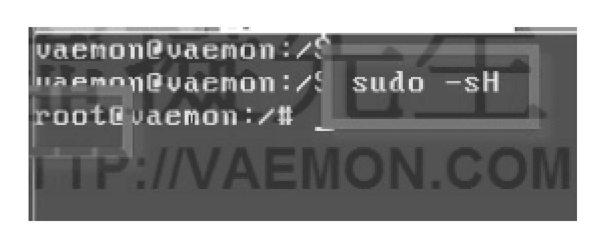
▲ 照着其步骤进行即可

▲ 系统会创建一个 Linux 下的 Cron Job 来 5 分钟运行一次

5. 配置 mrtg 的运行环境

接下来需要创建 mrtg 的运行环境。下面就是完整的步骤。

1. 首先我们以 root 的身份来运行，因为 mrtg 的配置文件是放在/etc 下，而/etc 一般是以 root 的身份运行以及管理的。键入 sudo -sH，自此以后都以 root 的身份来运行命令。
2. 进入之后，先创建 mrtg 的目录。在 Ubuntu 下，Apache2 的文档目录是安装在/var/www 中，因此我们要创建一个/var/www/mrtg。键入“mkdir /var/www/mrtg”。



▲ 先以 root 的身份运行较方便



▲ 创建一个目录来存放 mrtg 的配置文件

3. 接下来创建一个 mrtg.cfg 的配置文件。在 Linux 中，mrtg 的运行都是可读这个配置文件的内容，因此以后不管是 CPU、RAM 或是其他的监视条目，都是以这个文件为主。因此我们先创建，键入 cfgmaker public@localhost > /etc/mrtg.cfg。



▲ 创建配置文件

4. 接下来就要运行 mrtg 这个程序了。键入 env LANG=C mrtg /etc/mrtg.cfg，这个命令的目的是让 mrtg 可读/etc/mrtg.cfg 这个文件。

▲ 运行 mrtg 程序

5. 运行之后会有很多报警消息，多运行三、四次直到所有的报警消息都没有即可，在 Linux 下按上箭头就会重复运行上一个命令。

▲ 多运行几次报警消息会消失

6. 接下来就是创建 index.html 文件。也是键入 indexmaker /etc/mrtg.cfg > /var/www/mrtg/index.html。



▲ 创建网页的 index.html 文件

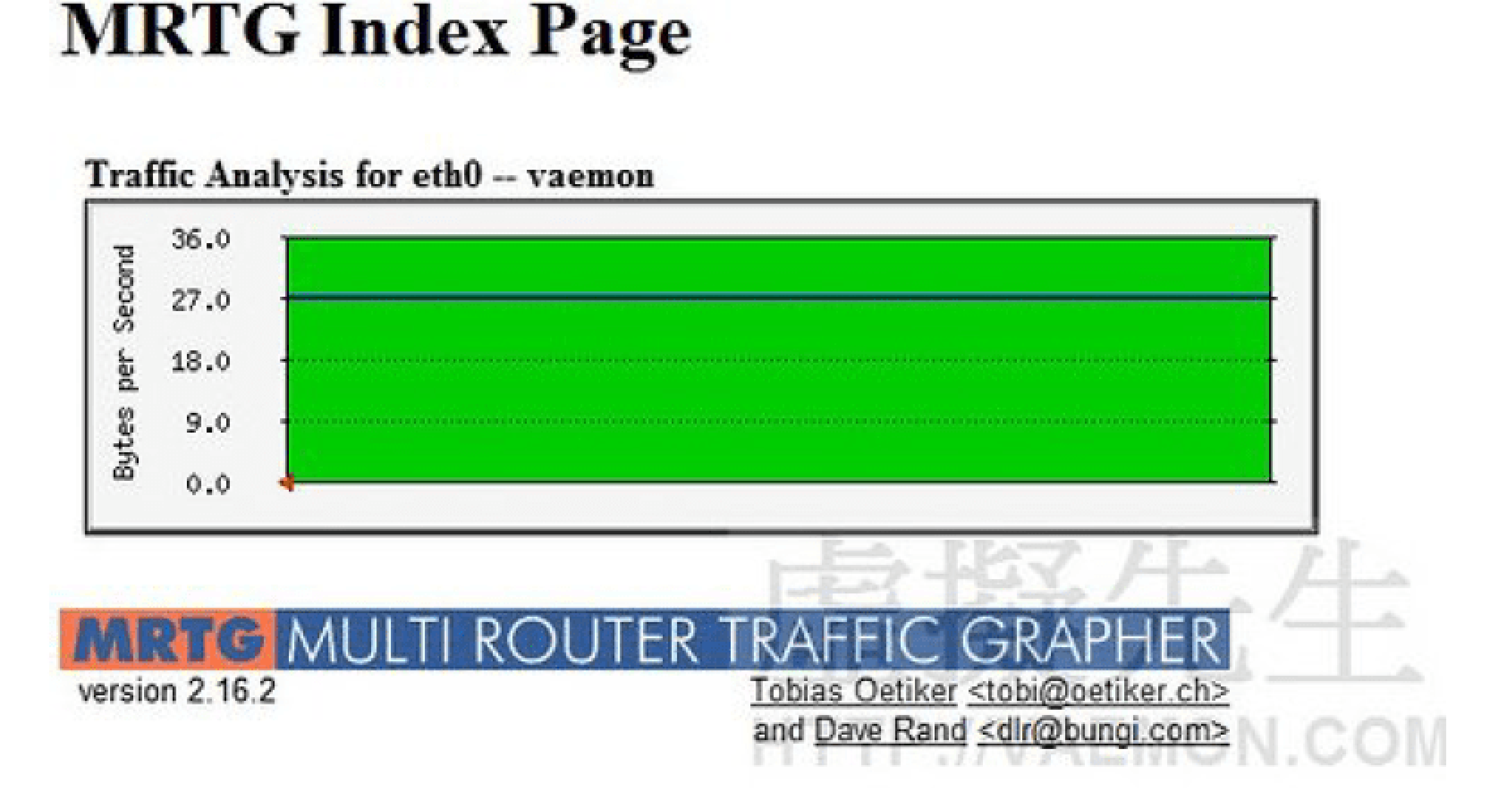
7. 我们可以到 mrtg 的目录下看到已经创建的图形及 html 文件。

```
aemon@vaemon:/var/www$ cd mrtg
aemon@vaemon:/var/www/mrtg$ ls -al
otal 152
rwxr-xr-x 2 root root  4096 2010-02-02 12:10 .
rwxr-xr-x 3 root root  4096 2010-01-28 13:41 ..
rw-r--r-- 1 root root  2418 2010-01-28 13:42 index.html
rw-r--r-- 1 root root  1768 2010-02-02 12:10 localhost_eth0-day.png
rw-r--r-- 1 root root  6592 2010-02-02 12:10 localhost_eth0.html
rw-r--r-- 1 root root 51151 2010-02-02 12:10 localhost_eth0.log
rw-r--r-- 1 root root  1725 2010-02-02 11:45 localhost_eth0-month.png
rw-r--r-- 1 root root 51151 2010-02-02 12:05 localhost_eth0.old
rw-r--r-- 1 root root  1802 2010-02-02 11:45 localhost_eth0-week.png
rw-r--r-- 1 root root  1910 2010-02-02 11:45 localhost_eth0-year.png
rw-r--r-- 1 root root   538 2010-01-28 13:41 mrtg-l.png
rw-r--r-- 1 root root   414 2010-01-28 13:41 mrtg-m.png
rw-r--r-- 1 root root  1759 2010-01-28 13:41 mrtg-r.png
aemon@vaemon:/var/www/mrtg$ _
```

▲ 可以看到文本和图档都已经创建了

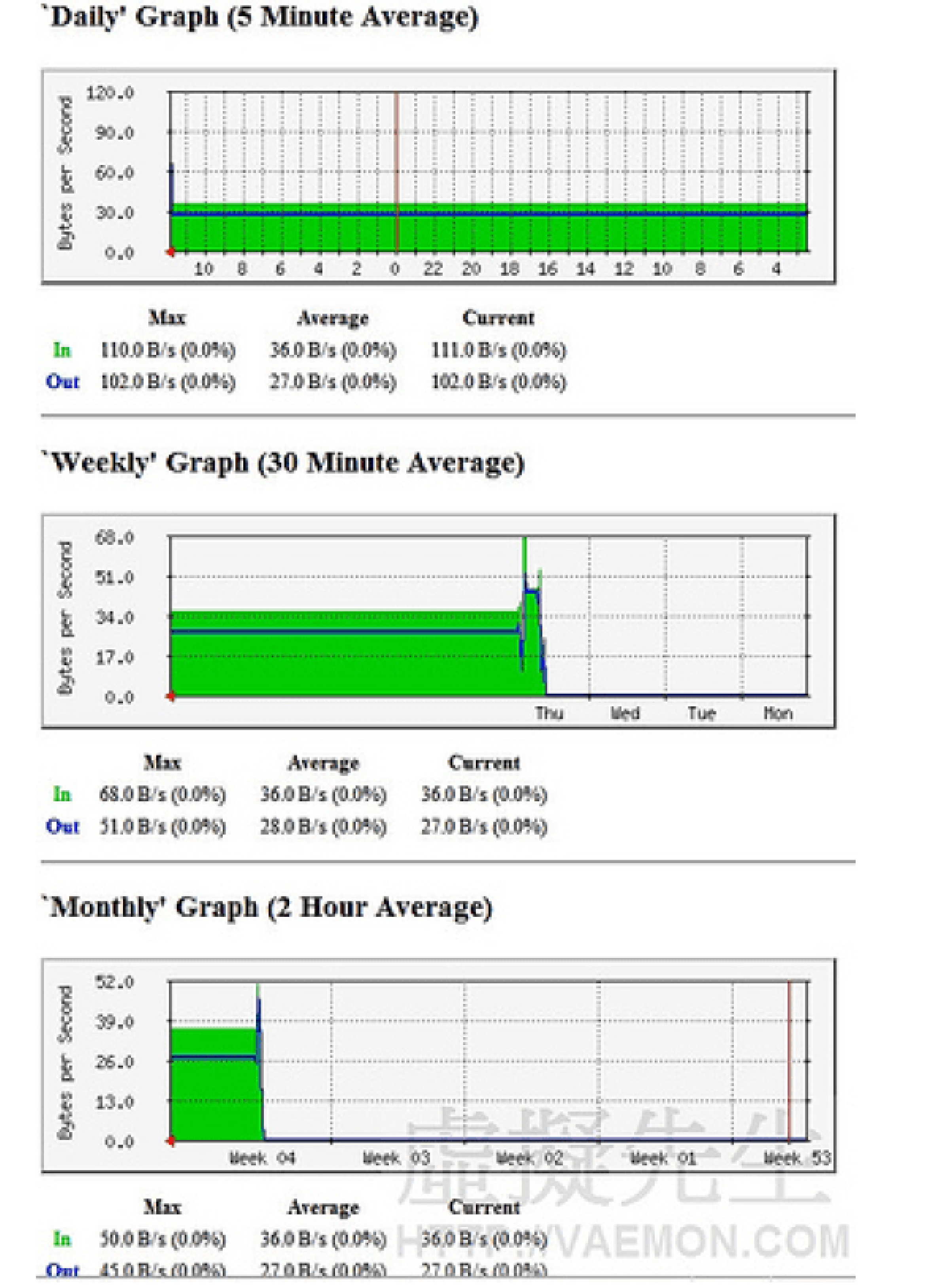
6. 从浏览器中观察 mrtg 的图表

我们从能访问该 Linux 主机上的计算机的浏览器进入，本例为 http://192.168.1.223/mrtg。就可以进入。进入后只有一个图表，单击可以看到更详细的图表。



▲ 键入网址

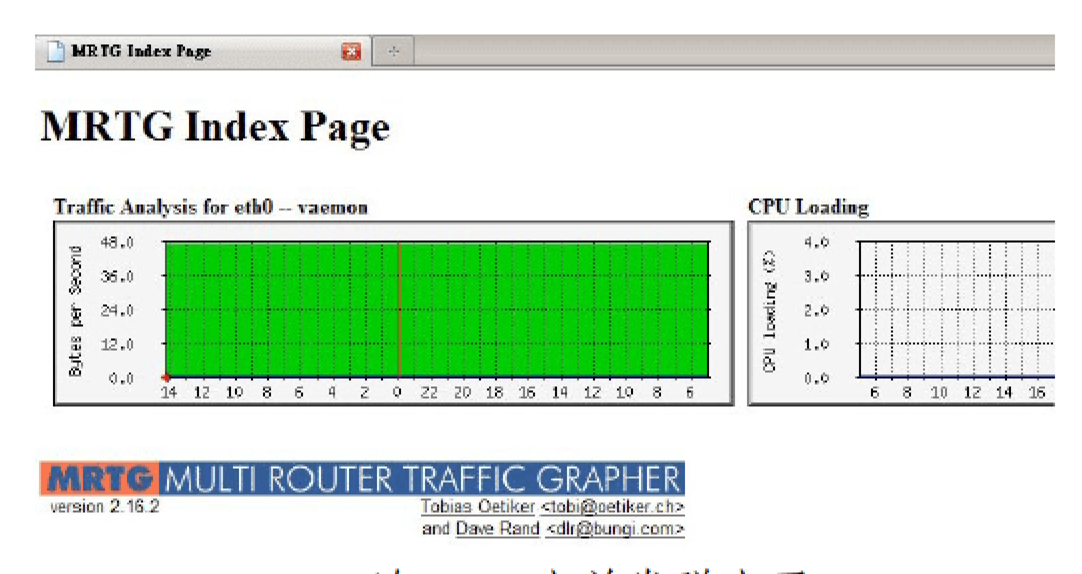
到此，我们的 mrtg 已经安装好了，如果我们想要监视其他的条目，只要更改/etc/mrtg.cfg，并且重新运行 mrtg，重新生成 index.html，就可以加入新的 mrtg 监视条目，只要将这里 mrtg 的功能安装好之后，以后就方便了。



▲ 会有不并发段的图档，对于长时间的资源使用率掌握很有帮助

6.2.3 加入 CPU 的监测

MRTG 安装好之后，Linux 已经具有监视 eth0 的功能了。当然要将 Linux 主机转换到虚拟机之前，你还得理解最重要的 CPU 等效率，在 P2V 之后才知道要如何增强这些效能。



▲ 连 CPU 也并发弹出了

▶ 安装 CPU 效能监测

要将 MRTG 拿来评估 CPU 效能，下面是必须运行的步骤。

（1）安装 sysstat CPU 监测程序。

（2）创建 CPU 监测运行脚本文件。

（3）更改脚本文件为可运行权限。

（4）更改 mrtg.cfg 运行环境参数档。

（5）以配置文件中的参数来运行 mrtg。

（6）创建网页 index.html。

（7）如果需要，可以安装烧机程序来烧 CPU。

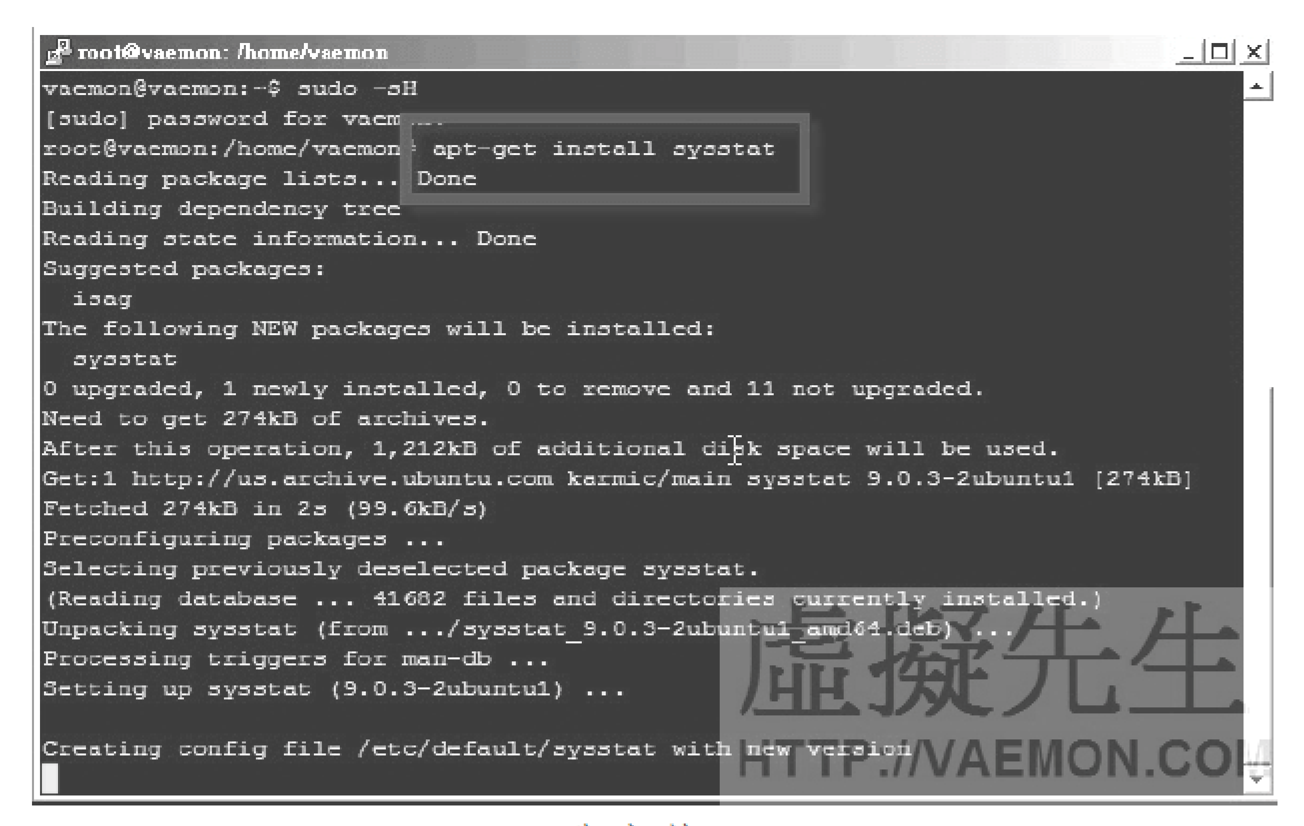
（8）从网页检视 mrtg 的图表。

1. 安装 sysstat CPU 监测程序

要监测 CPU，必须先安装一个 sysstat 中的 sar 来定时运行 CPU 的使用率（Utilization %）。只要你能连上网络，并且能使用 root 的权限运行程序，就可以用 apt-get 的方式来安装了。

▶ 安装 sysstat

1. 键入 sudo apt-get install sysstat，键入用户口令之会弹出安装的画面。
2. 安装时系统会要求键入 y，意味着同意安装，之后就会进行一些说明画面，并且在安装完毕时回到命令行提示下。



▲ 先安装 sysstat

3. 当安装完毕，我们可以用下面的命令检视是否安装成功，sar -u 2 5 意味着每 2 秒监视一次 CPU 的使用率，监测 5 次。



▲ 用这个程序来监视 CPU 的使用率

2. 创建 CPU 监测脚本文件

接下来创建监测 CPU 使用率的脚本文件，我们把这个脚本文件放在/opt/mrtg 之下的 mrtg.cpu，以后可以将内存、硬盘 IO 或其他要监测的都放在同一个目录下用不同的名称代替。

▶ 创建 CPU 监测脚本文件

1. 键入 sudo mkdir /opt/mrtg，创建目录。

2. 键入 vim /opt/mrtg/mrtg.cpu，编辑 CPU 的脚本文件。

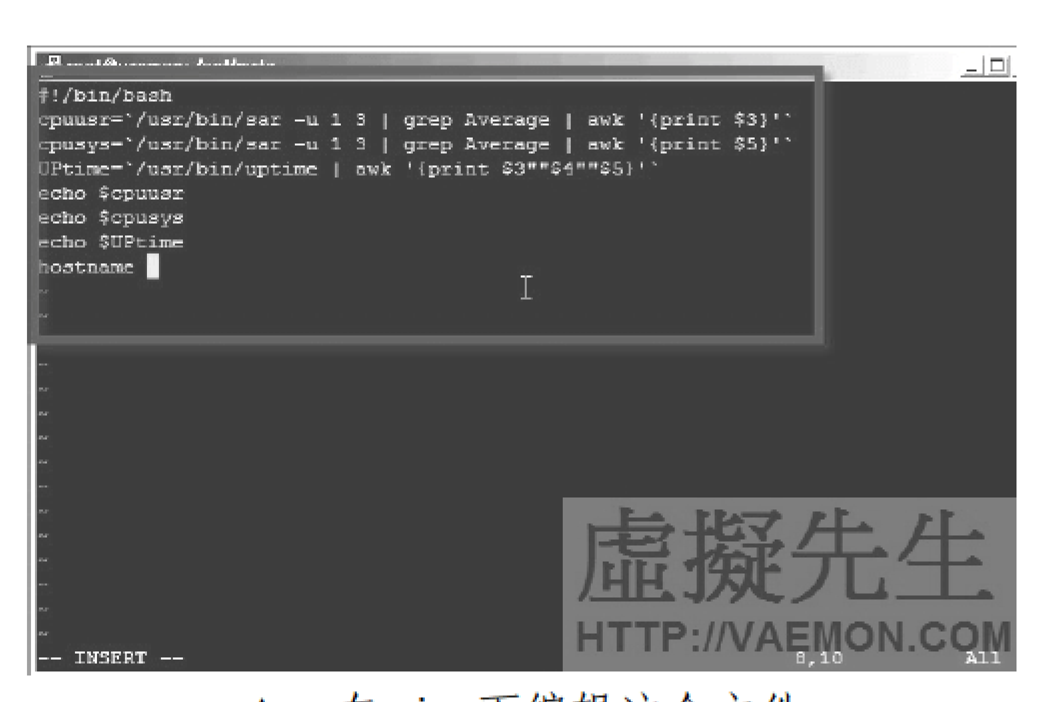


▲ 创建 mrtg.cpu 的运行档

3. 进入之后，将下列的代码键入到这个 mrtg.cpu 文件中：

```
#!/bin/bash
cpuusr=`/usr/bin/sar -u 1 3 | grep Average | awk '{print $3}'`
cpusys=`/usr/bin/sar -u 1 3 | grep Average | awk '{print $5}'`
UPtime=`/usr/bin/uptime | awk '{print $3"$4"$5}'`
echo $cpuusr
echo $cpusys
echo $UPtime
hostname
```

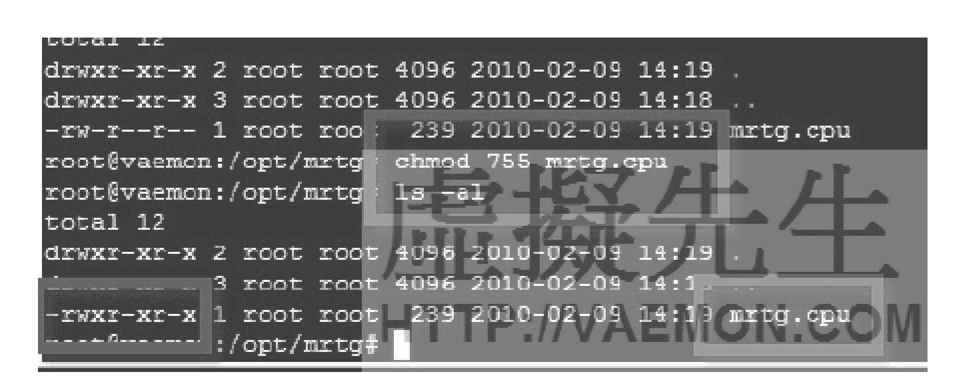
4. 键入完之后别忘了存盘离开 vim。



▲ 在 vim 下编辑这个文件

3. 将 mrtg.cpu 配置为可执行文件

在创建完 mrtg.cpu 之后，要将这个文件配置成可运行。我们键入 chmod 755 /opt/mrtg/mrtg.cpu 这个文件变成绿色之后，意味着可以运行了，或是可以检视其运行的权限是否有打开。



▲ 将 mrtg.cpu 变成可执行文件

4. 修改 mrtg 的运行参数档 mrtg.cfg

接下来我们要编修 mrtg.cfg 这个参数档，只要键入 sudo vim /etc/mrtg.cfg 即可。将下面的代

码贴到 mrtg.cfg 的最后面，注意千万不要动到 mrtg.cfg 的其他地方！

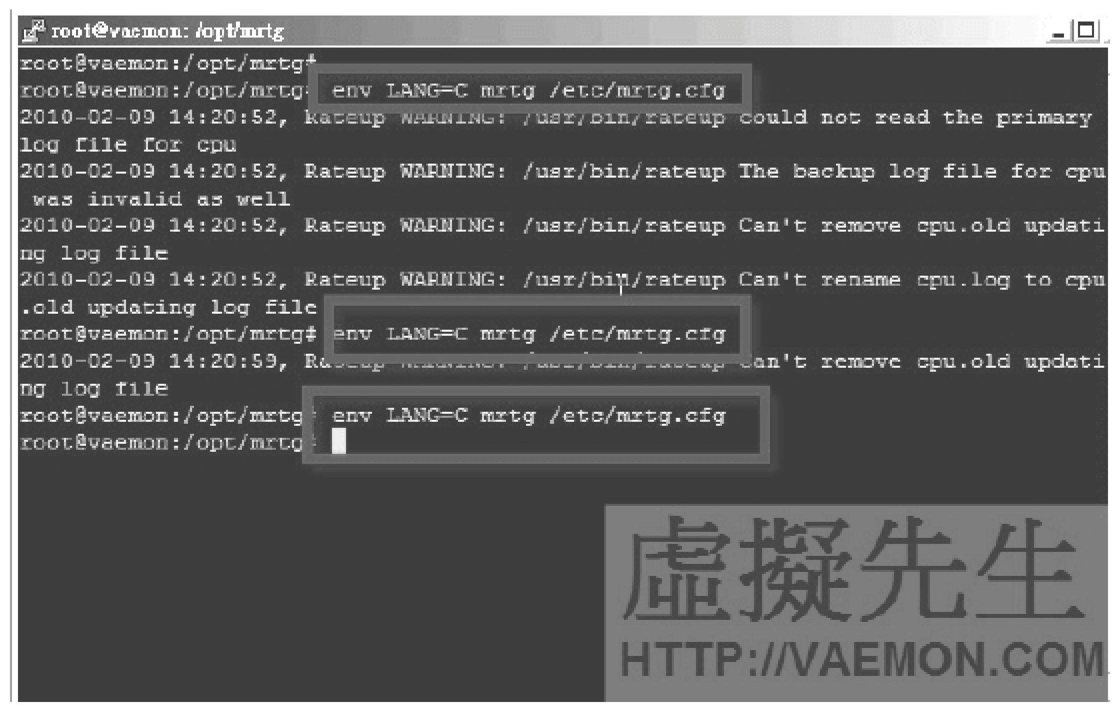
```
Target[cpu]: `/opt/mrtg/mrtg.cpu`
MaxBytes[cpu]: 100
Options[cpu]: gauge, nopercent, growright
YLegend[cpu]: CPU loading (%)
ShortLegend[cpu]: %
Legend0[cpu]:  CPU us;
LegendI[cpu]:  CPU sy;
Title[cpu]: CPU Loading
PageTop[cpu]:<H1>CPU Loading</H1>
```



▲ 修改 mrtg.cfg

5. 配置及创建 mrtg 的运行环境

接下来就要运行 mrtg 这个程序了。键入 env LANG=C mrtg /etc/mrtg.cfg，运行之后会有很多报警消息，多运行三、四次直到所有的报警消息都没有即可，在 Linux 下按上箭头就会重复运行上一个命令。



▲ 生成 CPU 的运行档之后，就要重新配置 mrtg 的配置文件

接着键入“indexmaker /etc/mrtg.cfg > /var/www/mrtg/index.html”。我们可以到 mrtg 的目录下看到已经创建的图形及 html 文件。

```
ng log file
root@vaemon:/opt/mrtg# env LANG=C mrtg /etc/mrtg.cfg
root@vaemon:/opt/mrtg# indexmaker /etc/mrtg.cfg > /var/www/mrtg/index.html
root@vaemon:/opt/mrtg# 
```


▲ 重新生成网页文件

```
root@vaemon:/var/www/mrtg#
root@vaemon:/var/www/mrtg# ls -al
total 272
drwxr-xr-x 2 root root  4096 2010-02-09 17:30 .
drwxr-xr-x 3 root root  4096 2010-01-28 13:41 ..
-rw-r--r-- 1 root root  1597 2010-02-09 17:30 cpu-day.png
-rw-r--r-- 1 root root  5890 2010-02-09 17:30 cpu.html
-rw-r--r-- 1 root root 48221 2010-02-09 17:30 cpu.log
-rw-r--r-- 1 root root  1406 2010-02-09 16:25 cpu-month.png
-rw-r--r-- 1 root root 48240 2010-02-09 17:25 cpu.old
-rw-r--r-- 1 root root  1518 2010-02-09 17:10 cpu-week.png
-rw-r--r-- 1 root root  1706 2010-02-09 14:20 cpu-year.png
-rw-r--r-- 1 root root  2612 2010-02-09 14:21 index.html
-rw-r--r-- 1 root root  1987 2010-02-09 17:30 localhost_eth0-day.png
-rw-r--r-- 1 root root  6582 2010-02-09 17:30 localhost_eth0.html
-rw-r--r-- 1 root root 50902 2010-02-09 17:30 localhost_eth0.log
-rw-r--r-- 1 root root  1942 2010-02-09 16:20 localhost_eth0-month.png
-rw-r--r-- 1 root root 50920 2010-02-09 17:25 localhost_eth0.old
-rw-r--r-- 1 root root  2027 2010-02-09 17:30 localhost_eth0-week.png
-rw-r--r-- 1 root root  2104 2010-02-09 14:20 localhost_eth0-year.png
-rw-r--r-- 1 root root   538 2010-01-28 13:41 mrtg-l.png
-rw-r--r-- 1 root root   414 2010-01-28 13:41 mrtg-m.png
-rw-r--r-- 1 root root  1759 2010-01-28 13:41 mrtg-r.png
root@vaemon:/var/www/mrtg# 
```


▲ 看到新创建的网页档了

6. 从浏览器中观察 mrtg 的图表

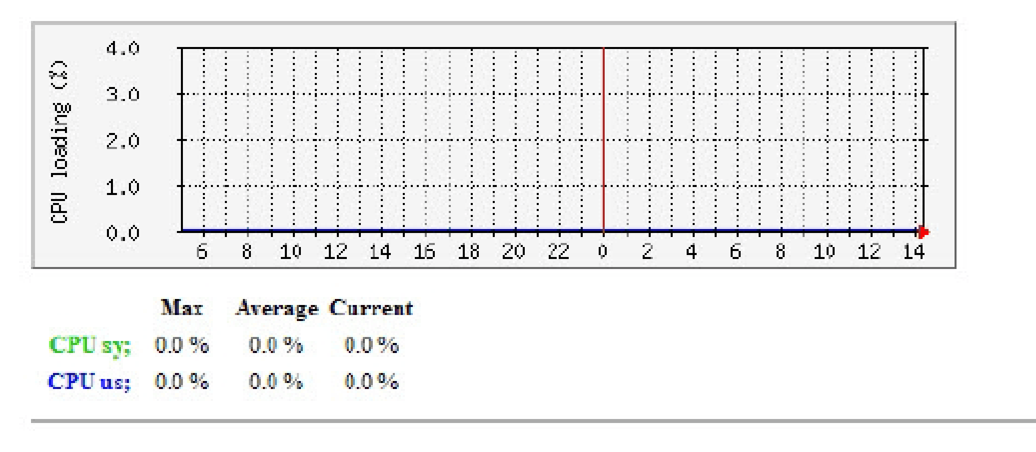
我们从能访问该 Linux 主机上的计算机的浏览器进入，本例为 http://192.168.1.223/mrtg。进入后变成两个图表了，左边的是原来的 eth0，右边的就是新的 CPU。单击可以看到更详细的图表。

到此，我们的 mrtg 已经安装好 eth0 和 CPU 了。如果我们想要监视其他的条目，只要更改 /etc/mrtg.cfg，并且重新运行 mrtg，并且重新生成 index.html，就可以加入新的 mrtg 监视条目，只要将这里 mrtg 的功能安装好，以后就方便了。

CPU Loading

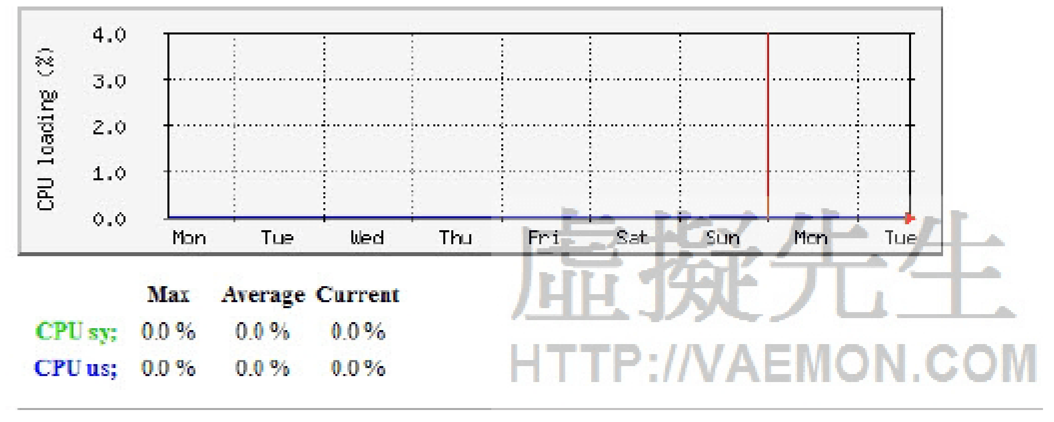
The statistics were last updated **Tuesday, 9 February 2010 at 14:21**, at which time '**vaemon**' had been up for **3min,1**.

`Daily' Graph (5 Minute Average)



	Max	Average	Current
CPU sy;	0.0 %	0.0 %	0.0 %
CPU us;	0.0 %	0.0 %	0.0 %

`Weekly' Graph (30 Minute Average)



	Max	Average	Current
CPU sy;	0.0 %	0.0 %	0.0 %
CPU us;	0.0 %	0.0 %	0.0 %

▲ 图为 CPU 的效能监视

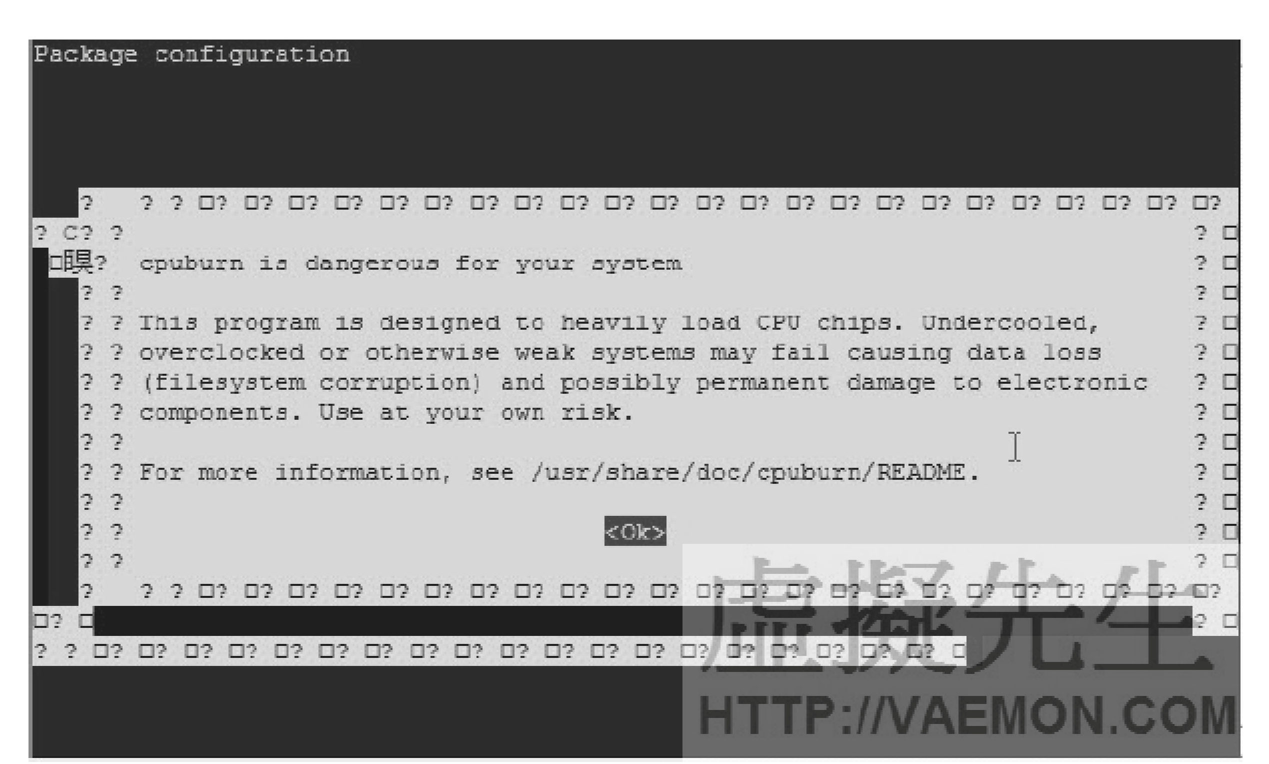
注意

安装 CPU 烧机程序

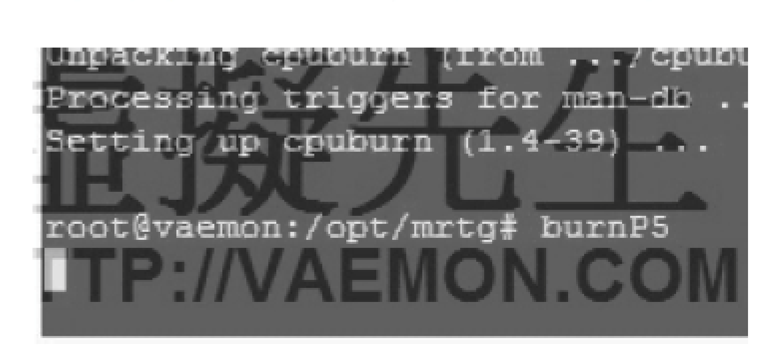
如果你想要安装个 CPU 烧机程序，可键入 sudo apt-get install cpuburn，就可以进行 CPU 的烧机，只要安装完之后键入： burnP5，就可以让你的 CPU 的 loading 冲到顶了。



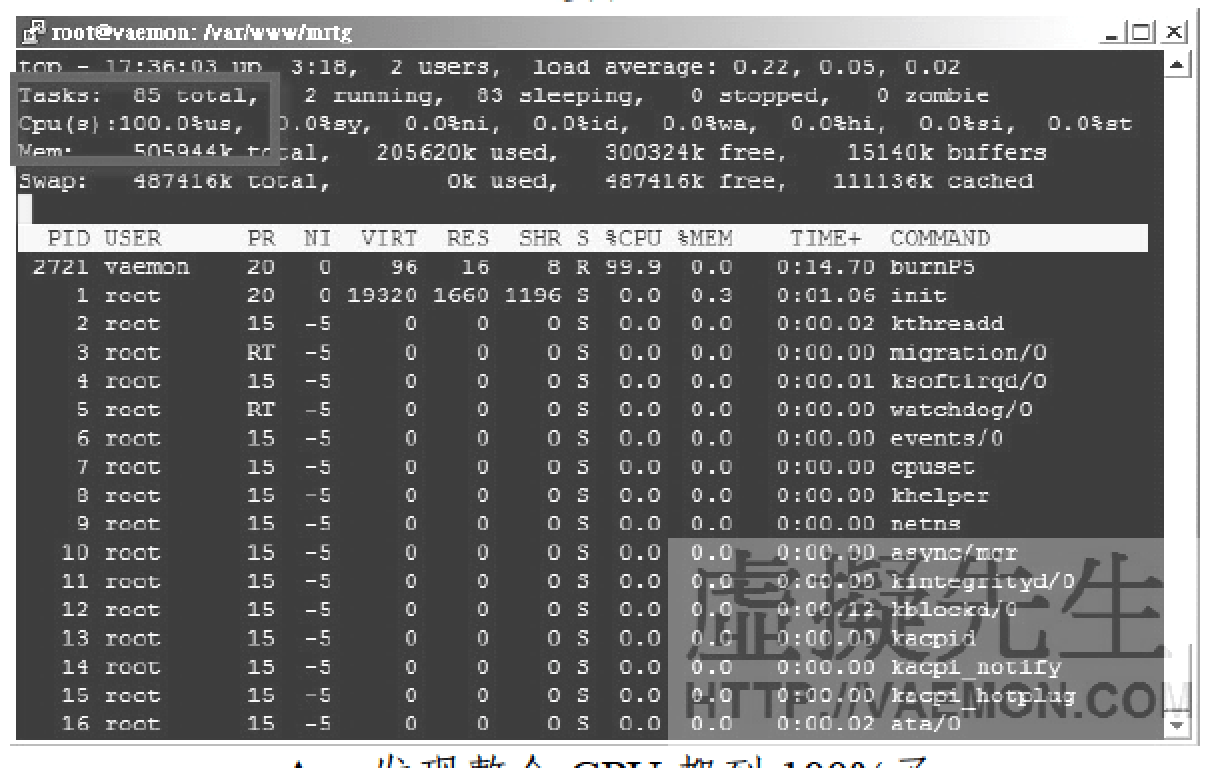
▲ 安装烧机程序



▲ 会有报警画面，因为有可能把系统弄蓝屏



▲ 运行 burnP5



▲ 发现整个 CPU 都到 100%了

结 语

vSphere 的对象本来就是 Windows 和 Linux（含 FreeBSD）的 X86 系统，而这两个系统的效能是虚拟化之前最需要理解的 ，以方便分配物理机的资源并决定虚拟化的合并率，更是针对虚拟机校调性能的最佳引用。

第 7 章 精算虚拟机的合并率、TCO 以及 ROI

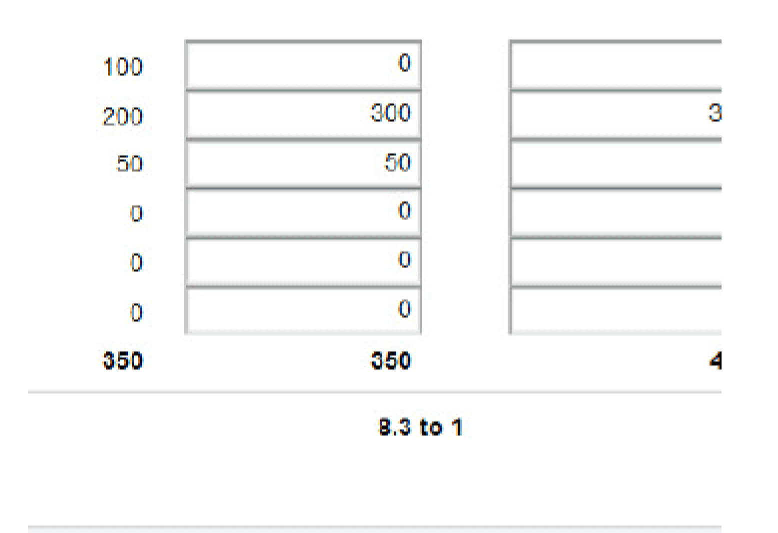
关键词：

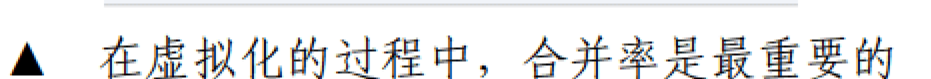
- 服务器的合并率计算
- 目标 CPU 使用率：不超过 90%
- SMP 及多核心的计算：每实体核心 3～5 个 vCPU
- 内存数量：2GB 到 4GB/每实体 CPU 核心
- 网卡的分配原则：5～7 个虚拟机/每网络界面
- 实体服务器配置：多台实体机或刀片服务器
- 桌面虚拟化的合并率计算
- CPU 计算：每实体核心 5～8 个 vCPU
- 内存数量：1GB/XP，2GB/Vista
- 网卡的分配原则：5～8 个虚拟机/每网络界面
- 实体服务器配置：多台服务器或多片刀片服务器
- 企业 IT 初始成本
- 设备购买
- 机房空间成本
- 软件及授权
- IT 营运成本
- 管理成本
- 折旧成本
- 设备的电费
- 服务器的租金
- 冷却费用
- 虚拟化仍有其他支出
- 服务器合并 TCO/ROI 计算
- 了解 VMware 的 ROI 工具
- 服务器集成的 ROI/TCO 计算
- TCO/ROI 计算器详细说明
- 金额说明
- 分类细部计算
- 虚拟桌面管理成本计算
- 决定需要合并的个人计算机数量
- 最花钱的部分为服务器及 VDI 软件

每一个计划虚拟化的企业都有几个问题，首先就是“一台实体机能装几台虚拟机？”，第二就是“虚拟化能帮我省多少钱？，第三就是“需要再花什么钱？”，第四就是“还需要什么其他的设备？”虽然这些问题的答案因不同的需求及行业类别、企业规模、虚拟机的合并率等而异，但虚拟化已经有几年的历史，业界也有了一致的标准，本章即以这些标准为例子帮助企业进行虚拟化的评估。

7.1 合并率的精算

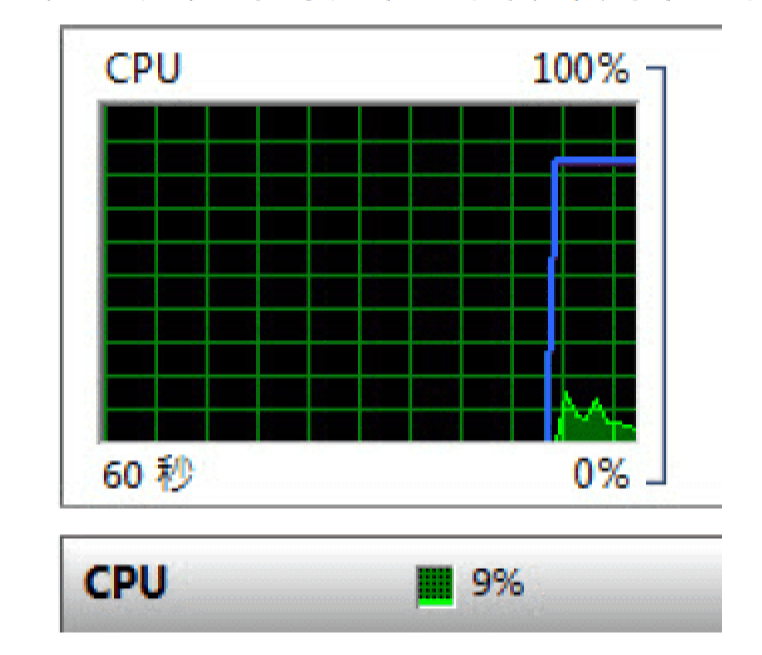
从成本的角度来看，虚拟化的确能帮企业省下大笔的花费。然而虚拟化并不是无限压缩的过程，而是一个“合理”的压缩过程。所谓的“合理”压缩，就是要让整个企业的所有服务都能正常运行的情况下来进行服务器的合并或是桌面机的集中。但这里的“合理”压缩并不容易掌握，在这一章中，我们主要是学习如何找出最“合理”的压缩值。





▲ 在虚拟化的过程中，合并率是最重要的

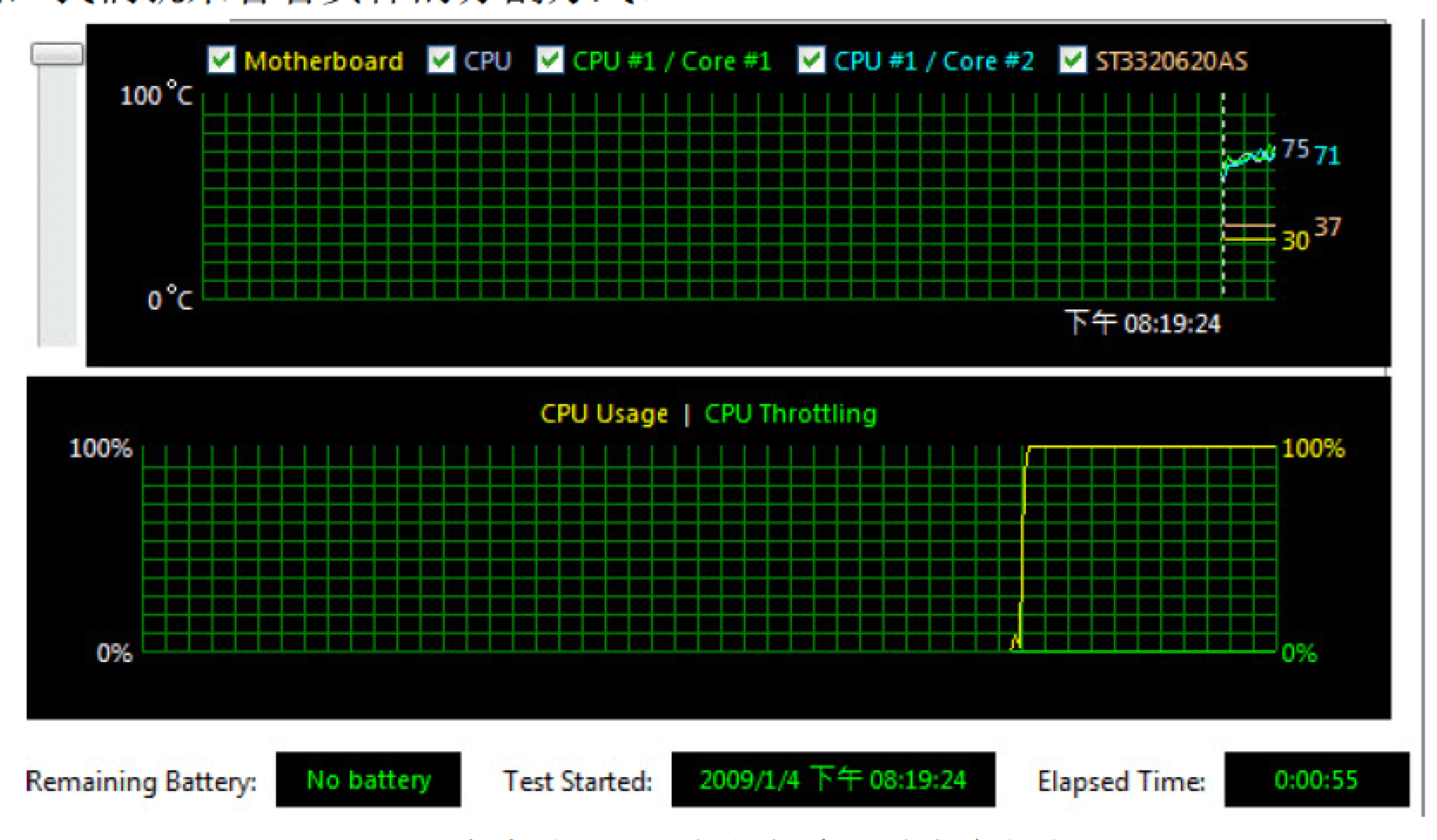
在虚拟化的过程中，“一台实体机上能放上几个虚拟机”是必须经过规划的。这个规划的过程称为 Provisioning。Provisioning 的过程绝不简单，通常会花上长时间的记录和分析。上一章我们曾经针对 CPU 进行了查看，可以得知目前实体机的 CPU 使用率。CPU 的使用率是虚拟机合并时最重要的依据，当然前提是存储设备已经独立出来了。当其他因素都排除之后，接下来就是不同场合的合并率计算了。



▲ 以 CPU 的使用率来分析合并率是最常用的方式

7.1.1 服务器的合并率计算

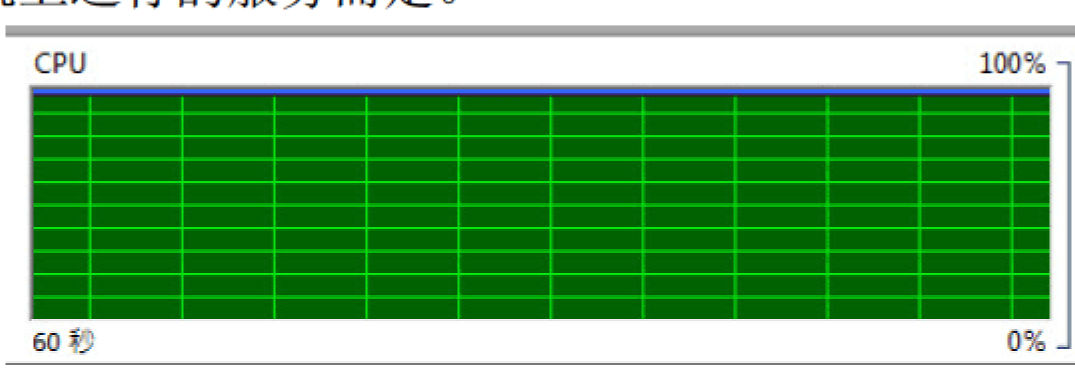
服务器提供了网络的服务，因此大部分的负担会在网络的传送上。有些消耗 CPU 的服务如数据库，是不建议放上虚拟机的。当然其他服务如网页（IIS 或 Apache）、应用程序服务（CGI、Perl 或是.NET）、邮件服务、FTP 服务等。由于主要的工作都在存储和网络之间，因此是最适合合并的场景，我们就来看看实体的分割方式。



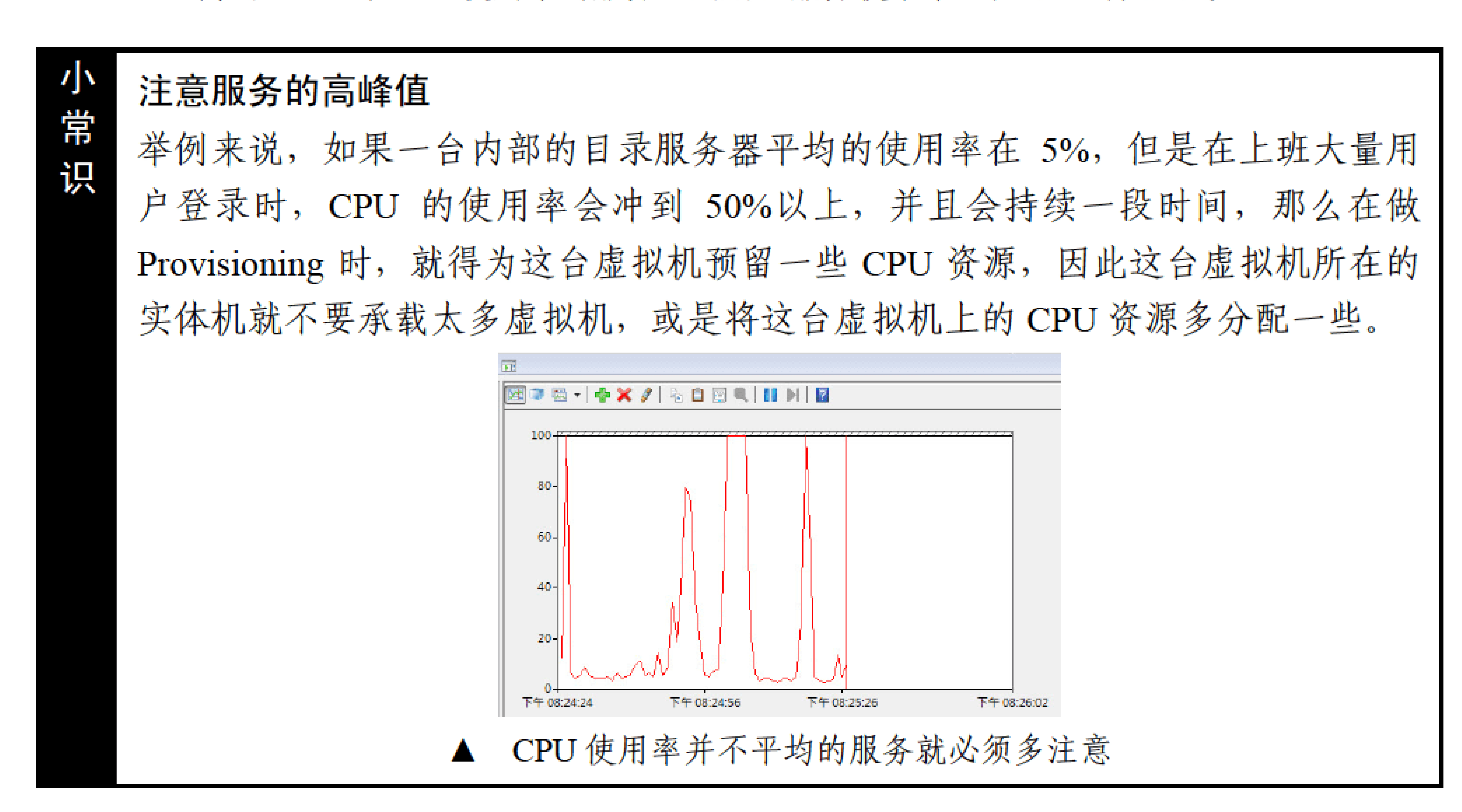
▲ 高负载 CPU 的服务并不适合虚拟化

1. 目标 CPU 使用率：不超过 90%

一般来说，CPU 的使用率在虚拟机上也是可以累加的。一台长年以 5%使用的实体机，如果放上 10 台类似的虚拟机，实体机的 CPU 使用率将落在 55%～60%，其中多出来的使用率视使用的虚拟机产品 Hypervisor 占用的资源而定。如果想要达到 90%的合并率，理想的虚拟机是 15 个，当然这还要视每个虚拟机上运行的服务而定。



▲ 高达 100%的 CPU 使用率的服务一定会让服务慢下来，因此不可太超过



小常识

注意服务的高峰值

举例来说，如果一台内部的目录服务器平均的使用率在 5%，但是在上班大量用户登录时，CPU 的使用率会冲到 50%以上，并且会持续一段时间，那么在做 Provisioning 时，就得为这台虚拟机预留一些 CPU 资源，因此这台虚拟机所在的实体机就不要承载太多虚拟机，或是将这台虚拟机上的 CPU 资源多分配一些。

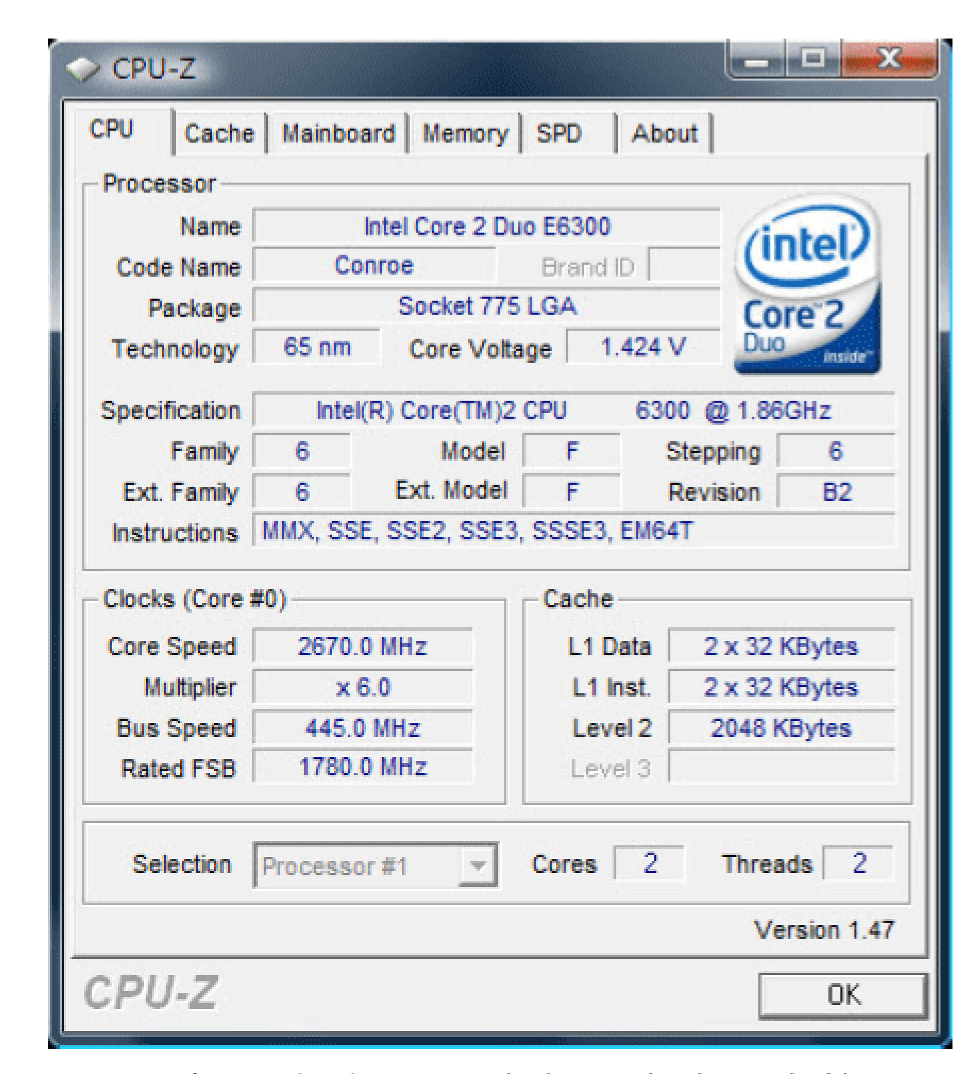
▲ CPU 使用率并不平均的服务就必须多注意

2. SMP 及多核心的计算：每实体核心 3～5 个 vCPU

在 2005 年之后 Intel/AMD 的 CPU 产品都支持多核心，一个核心就相当于 1 个实体的 CPU（pCPU），让 SMP/多核心的单一实体机上能承载的虚拟机数量再度加倍。由于在虚拟机产品中，所有的 CPU 是以 vCPU 来分配的，一个虚拟机可以不止有一个 vCPU（虚拟机多核心），因此一 SMP 且多核心的实体机，能承载的虚拟机数量当然不同。我们就举一个例子来看看。

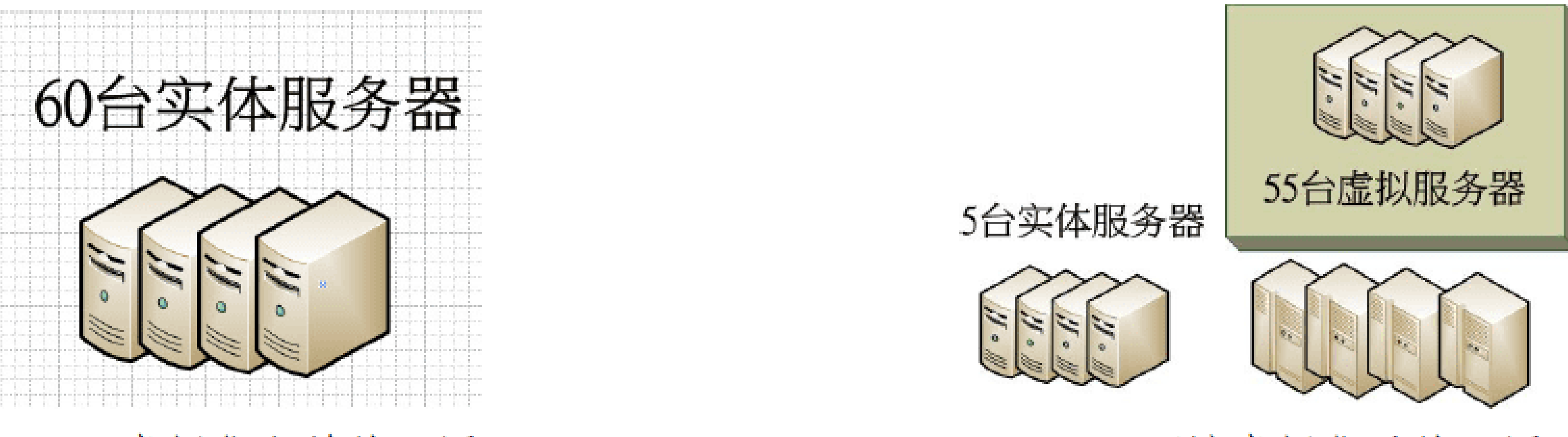
3. 企业的实体及虚拟 CPU 计算

- 企业中有 60 台实体机服务器
- 其中有 5 台是高负载的服务器，如 SQL 或是运算用
- 55 台是可以虚拟化的服务器，如网页或是其它服务



▲ 新一代的 CPU 上都至少有 2 个核心

- 有 45 台虚拟机的负载较小，每台使用 1 个 vCPU
- 有 10 台虚拟机的负载较大，每台使用 2 个 vCPU



▲ 虚拟化之前的配置　　▲ 预计虚拟化后的配置

从以上的 CPU 需求得知，我们需要的实体 CPU 核心（pCPU）数量如下：

```
(45 × 1 Vcpu) + (10 × 2 vCPU) = 45 vCPU + 20 vCPU = 65 vCPU's
```

前面说过，一个实体核心就算一个 pCPU。如果不想实体机的负载太高（超过 90%），就让一个 pCPU 只需承载 3 个 vCPU，我们所需要的实体 CPU 核心数量就是：

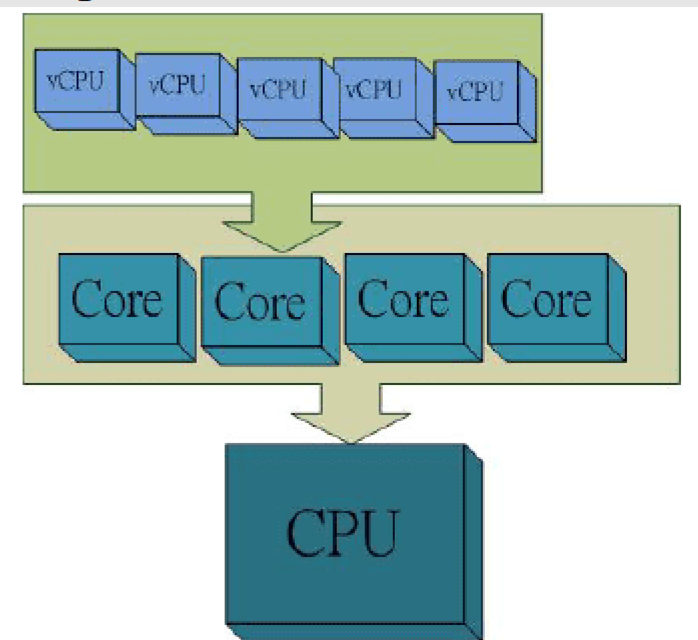
```
65 vCPU / 3 = 22 实体 CPU 核心
```

如果我们希望些实体机的合并率再高一点，那么就让一个 pCPU 承载 5 个 vCPU，我们所需要的实体 CPU 核心数量就是：

```
65 vCPU / 5 = 13 实体 CPU 核心
```

目前的服务器市场使用的产品大多是 4 核心的 CPU，因此在上面的例子中，需要的实体 4 核心 CPU 数量为：

```
22 Core / 4 = 6 CPU Packages 或是 13 Core /4 = 4 CPU Packages
```



▲ 需要的虚拟 CPU、实体 CPU 核心数和真正 CPU 个数，以 4 核心为例

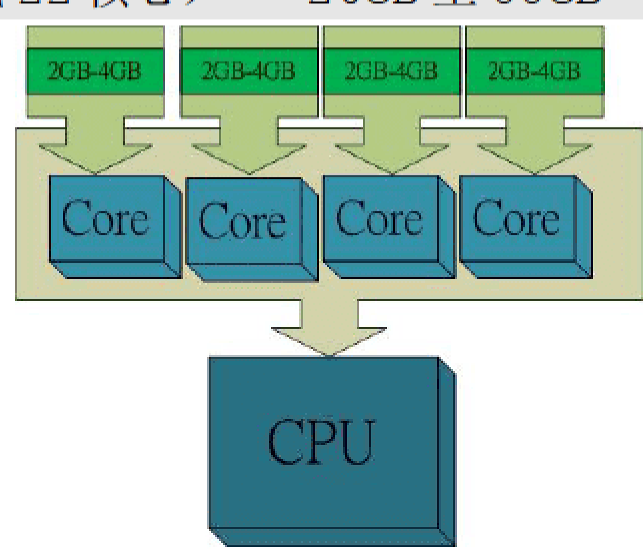
4. 内存数量：2GB 到 4GB/每实体 CPU 核心

视服务而定，不同的服务需要不同数量的内存，但是在服务器合并的场景中，大部分的服务都是网络上的功能，我们必须预先想到多人使用的场景，因此配备 2GB 的内存是最基本的，当然在服务较繁忙的 64 位服务器上，有些服务会将大部分的数据放上高速缓存中，因此需要更多的内存来满足需求，4GB 的内存当然也是主流的数量。

服务器的主板通常能插上 32GB 以上的内存，不过必须满足我们的基本要求，就是每一实体 CPU 对应到 2GB 到 4GB 的内存。如此一来，如果有一台 4 SMP 的主板，上面最大的实体 CPU 数量可能会到 4×4=16 个，那么所需的内存数量将会是 32GB 到 64GB，因此一台主机可能就不够，而必须有两台主机来承载这 16 个核心所衍生出来虚拟机了。

从上面的例子我们得知，这个系统需要的内存数量是：

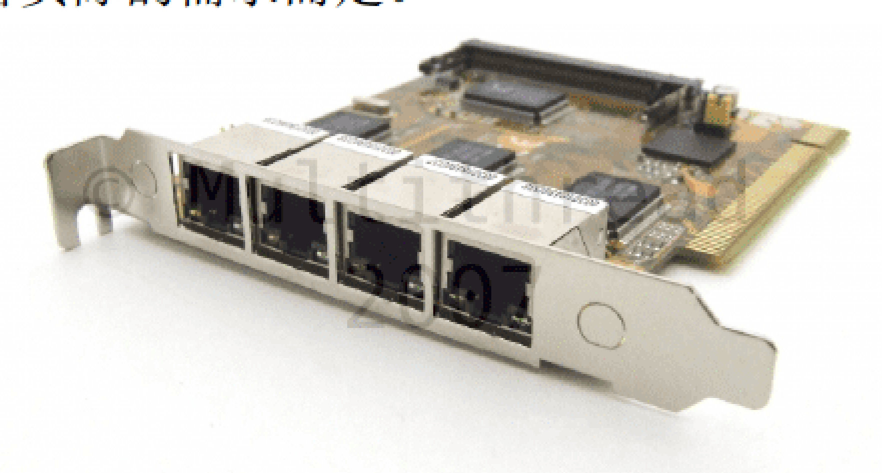
```
（2GB 或 4GB） × （13 核心或 22 核心） = 26GB 至 88GB
```



▲ 每一实体核心对应 2GB 到 4GB 的内存

5. 网卡的分配原则：5～7 个虚拟机/每网络界面

一台实体机上能安装的网络界面数量受限于 PCI 或 PCI-E/PCI-X 的数量。大部分的服务器已内置 2～3 个 GB 级网络界面，市面上也有单卡四界面（4 Ports）的网卡，因此一台服务器上能接上的网络界面最多可以到 20 个以上。此外，网络的整体速度和交换机也有关系，因此一个虚拟机需要几个网络界面，还是要看实际的需求而定。



▲ 真正需要时可以使用这种一卡四界面的产品

虽然目前的 GB 级网卡已成为主流，除了高负载的服务器之外，很少有机会将网络的带宽占满，而高负载的服务器又不适合虚拟化，因此一般在虚拟机上，我们会将一个网络界面分配给 3～5 个虚拟机使用，当然如果有需要单独实体网卡的服务，就不能和其他的虚拟机共享实体网卡了。从上面的例子，一共有 55 个虚拟机，因此需要的网络界面为：

```
55 个虚拟机 / （一个网络界面为 5～7 个虚拟机） = 8～11 个网络界面
```

6. 实体服务器配置：多台实体机或刀片服务器

从以上的计算，我们可以得知本例中的实体服务器以及硬件配置如下。

▶ 实体服务器配置

- 虚拟机数目：55 个
- CPU 核心数：13 或 22 核心
- 实体 CPU 数量：4 或 6 个 4 核心 CPU（使用 Xeon 53XX 或 74XX 家族）
- 实际内存数量：26GB 或 88GB
- 实体网络界面：8～11 个网络界面

由以上的数据看来，可以归纳出下面的最小实体服务器需求：

7. 最小实体服务器配置

- CPU 配置：2 个 4 核 CPU（共 8 核心）
- 内存：64GB（每 pCPU 分配 4GB，剩下的给自己用）
- 网卡：内置 2 个网络界面+1 张 2 Port 的网卡

- 每台虚拟机量：28 个（23 个 1vCPU 及 5 个 2 vCPU）
- 机器数量：2 台（共有 56 台虚拟机）

如果要应付较高的负担，则可以考虑下面的配置：

8. 较大负担的实体服务器配置

- CPU 配置：刀片服务器 2 个 4 核心或 6 核心（共 8 或 12 核心）
- 内存：64GB（每 pCPU 分配 4GB，剩下的给自己用）
- 网卡：使用刀片的 Network Bus 以及 Module 扩充
- 每台虚拟机量：18 个（15 个 1vCPU 及 3 个 2 vCPU）
- 机器数量：3 片刀片 （共有 54 台虚拟机）

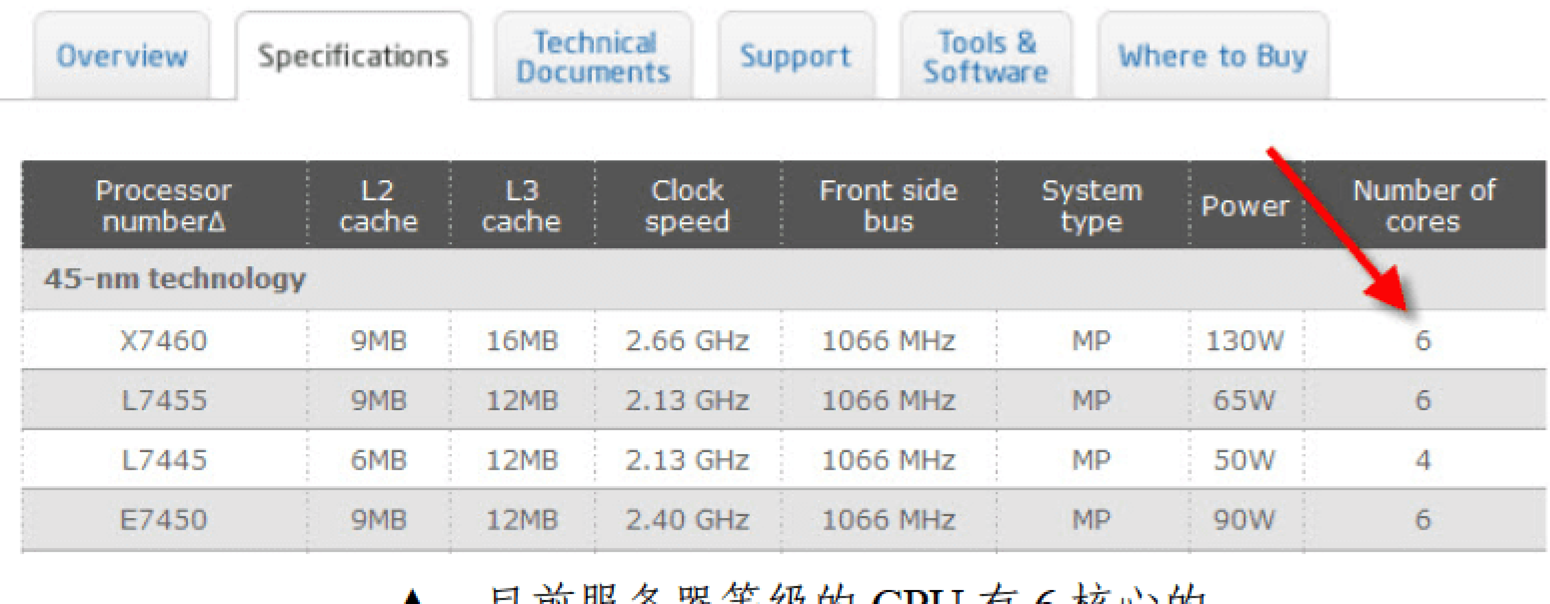


▲ 刀片服务器可以插入不同的模块来扩充功能

小常识

CPU 品牌的选择

虽然 1 个实体 CPU 核心可以承载 3～5 个 vCPU，但不同品牌/频率的 CPU 效能上存在差异，会让系统管理人员担心不同厂牌的产品造成性能上的差异。企业虚拟化大多使用 Intel Xeon 或 AMD Opteron 两种服务器专用 CPU，桌面系统的 CPU（如 Intel 的 Core i7 或是 AMD 的 Phenom）并不适合。此外 Xeon 或 Opteron 虽然有许多等级，但结构上的差异并不大。以 Cost/Performance 的角度来看，建议使用最高等级之下 1 级或 2 级的产品，如 Xeon 最高等级是 X7460，但较慢频率的 E7450 不但便宜很多，而且架构完全一样，也是 6 核心，可说是最平衡的选择。



Intel® Xeon® Processor 7000 Sequence
Specifications

Overview | Specifications | Technical Documents | Support | Tools & Software | Where to Buy

Processor numberΔ	L2 cache	L3 cache	Clock speed	Front side bus	System type	Power	Number of cores
45-nm technology							
X7460	9MB	16MB	2.66 GHz	1066 MHz	MP	130W	6
L7455	9MB	12MB	2.13 GHz	1066 MHz	MP	65W	6
L7445	6MB	12MB	2.13 GHz	1066 MHz	MP	50W	4
E7450	9MB	12MB	2.40 GHz	1066 MHz	MP	90W	6

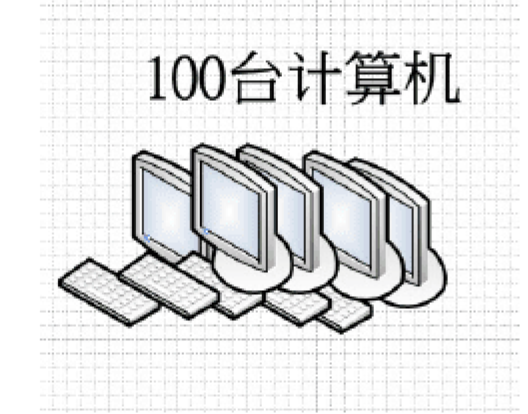
▲ 目前服务器等级的 CPU 有 6 核心的

7.1.2 桌面虚拟化的合并率计算

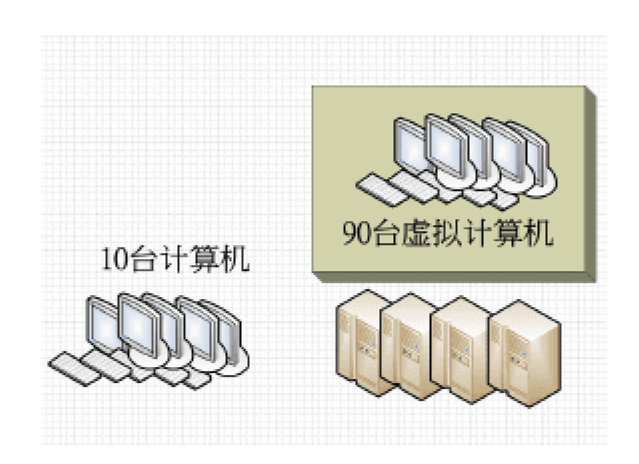
桌面虚拟化的合并率和服务器大同小异，计算方式也较单纯，在应用上较着重于内存的直接对应关系。

1. 桌面虚拟化实体及虚拟 CPU 计算

- 企业中有 100 台实体机桌面机器；
- 其中有 10 台是必须使用独立的显示适配器，不适合虚拟化；
- 90 台是可以虚拟化的日常工作个人计算机；
- 60 台使用 1GB 的 XP 操作系统；
- 30 台使用 2GB 的 Vista 操作系统。



▲ 桌面虚拟化之前



▲ 预计的桌面虚拟化之后

2. CPU 计算：每实体核心 5～8 个 vCPU

桌面系统使用的工作较为单纯，主要为办公室应用以及网页浏览，因此我们可以让实体 CPU 多负担一些。

从以上的 CPU 需求得知，我们需要的实体 CPU 核心（pCPU）数量如下：

```
(90 × 1 Vcpu) = 90 vCPU's
```

一个实体核心承载 5 个 vCPU，我们所需要的实体 CPU 核心数量就是：

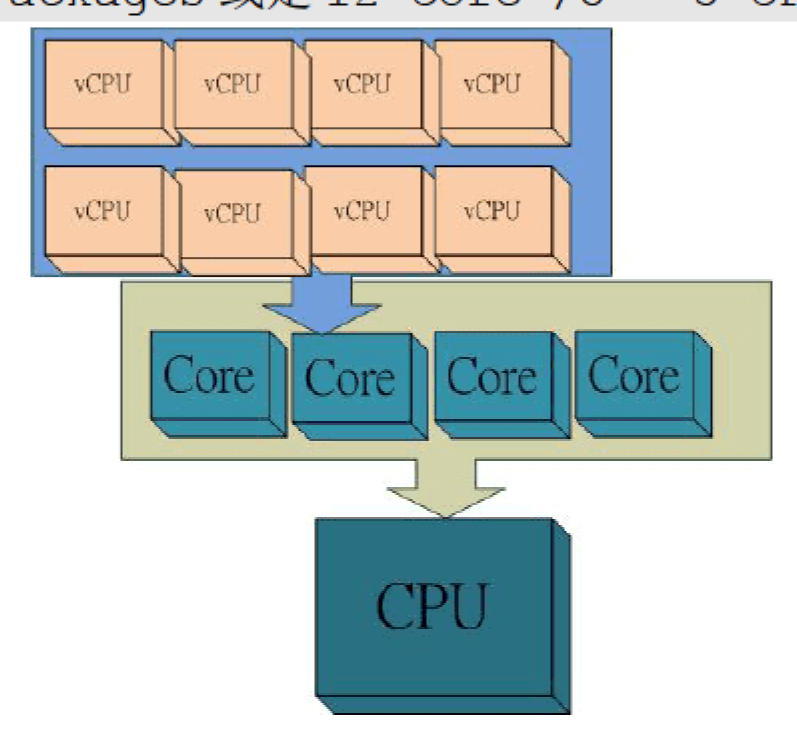
```
90 vCPU /5 = 18 实体 CPU 核心
```

如果我们希望些实体机的合并率再高一点，那么就让一个 pCPU 承载 8 个 vCPU，我们所需要的实体 CPU 核心数量就是：

```
90 vCPU /8 = 12 实体 CPU 核心
```

目前的服务器市场使用的产品大多是 4 核心的 CPU，因此在上面的例子中，需要的实体 4 核心 CPU 数量为：

```
18 Core / 4 = 5 CPU Packages 或是 12 Core /5 = 3 CPU Packages
```

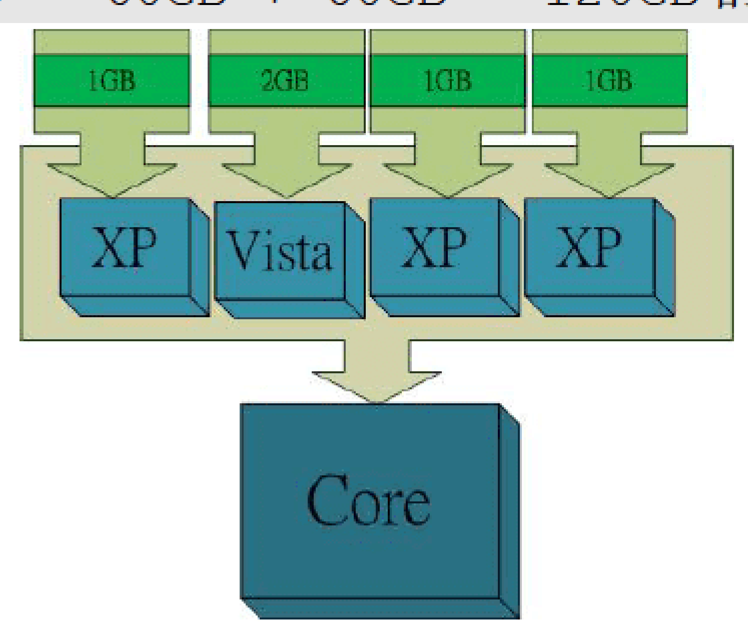


▲ 桌面虚拟化的 CPU 数量指示图

3. 内存数量：1GB/XP，2GB/Vista

桌面系统使用 1GB 的内存已足够，就算是 Vista Business 也不会造成“瓶颈”，但分配 2GB 给使用 Vista 的桌面系统，因此实际需要的内存数目为：

```
1GB × 60 + 2GB × 30 = 60GB + 60GB = 120GB 的内存
```



▲ 使用 Vista 的系统要多一点内存

4. 网卡的分配原则：5～8 个虚拟机/每网络界面

大部分的桌面用户主要是使用 RDP 或是 ICA 的远程桌面来传送画面，在 LAN 内 100MB 的速度已足够，因此我们可以让一个网络界面上传送 5～8 个虚拟机，因此需要的网络界面为：

```
90 个虚拟机 / （一个网络界面为 5～8 个虚拟机） =12～18 个网络界面
```

5. 实体服务器配置：多台服务器或多片刀片服务器

从以上的计算，我们可以得知本例中的实体服务器以及硬件配置如下：

- 虚拟机数目：90 个
- CPU 核心数：12 或 18 核心
- 实体 CPU 数量：3 或 5 个四核心 CPU（使用 Xeon 53XX 或 74XX 家族）
- 虚拟机总内存：120GB
- 实体网络界面：12～16 个网络界面

由以上的数据看来，可以归纳出下面的最小实体服务器需求。

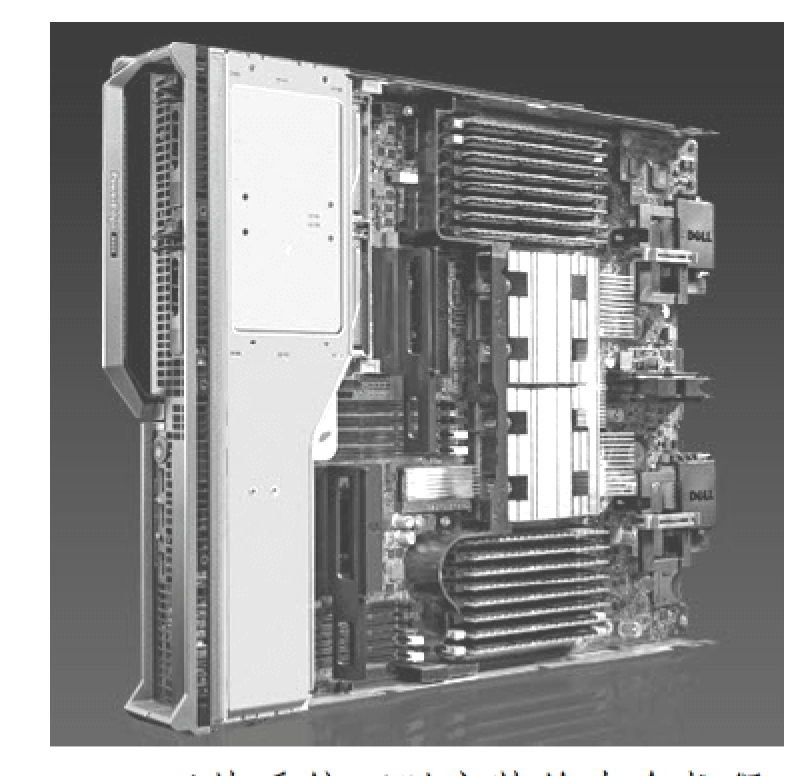
6. 最小单台实体服务器配置

- CPU 配置：1 个四核心或 2 个双核心 CPU（共四核心）
- 内存：32GB（29GB 给虚拟机，剩下的给自己使用）
- 网卡：内置 2 个网络界面+1 张网卡 （共 3 个网络界面）
- 每台虚拟机数：17 台 XP 及 6 台 Vista（共 23 台虚拟机）
- 机器数量：4 台（可承载 92 台虚拟机）

如果要应付较高的负担，则可以考虑下面的配置。

7. 较大负担的单台实体服务器配置

- CPU 配置：刀片服务器 2 个四核心 SMP（共八核心）
- 内存：48GB（40GB 给虚拟机，剩下的自己用）
- 网卡：使用刀片的 Network Bus 以及 Module 扩充
- 每台虚拟机数：20 台 XP 及 10 台 Vista（共 30 台虚拟机）
- 机器数量：3 片刀片（可承载 90 台虚拟机）



▲ 刀片系统可以安装的内存都很大，如 Dell M900 可装上 128GB

> **小常识**
>
> **桌面虚拟化 CPU 品牌的选择**
>
> 虽然是桌面虚拟化，但在服务器端的 CPU 需求也是相当频繁的。由于 Xeon 或 Opteron 的 CPU 最适合虚拟化的场合，因此就算是应用在桌面虚拟化的场合，承载虚拟机的实体服务器还是使用服务器产品的 CPU 而非桌面产品的 CPU。

7.2 成本的精算

在回答了第一个问题“一台实体服务器能承载几台虚拟机”之后，接下来的问题就是和成本及预算有关了。“虚拟化能帮我省多少钱？”“需要再花什么钱？”“还需要什么其他的设备？”在这一小节中，我们将给您标准答案！

7.2.1 企业的成本结构

不管是服务器合并或是虚拟桌面管理，最大的目的就是节省成本。这里所指的成本，在 IT 部分主要分为两大类，包括了公司成立时的初始花费以及营运时的花费。

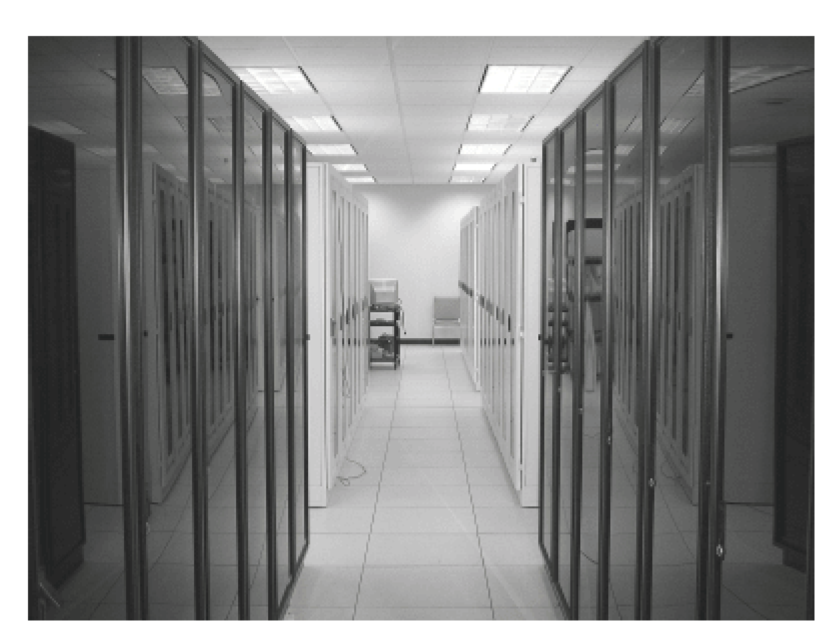
1. 无虚拟化时的成本

就算没有虚拟化，企业仍然会有 IT 成本的，分别为初始成本及营运成本。

2. 企业 IT 初始成本

- 设备购买
- 机房空间成本
- 软件及授权

以及新公司和营运中公司都必须面对的营运成本，其中包括了：



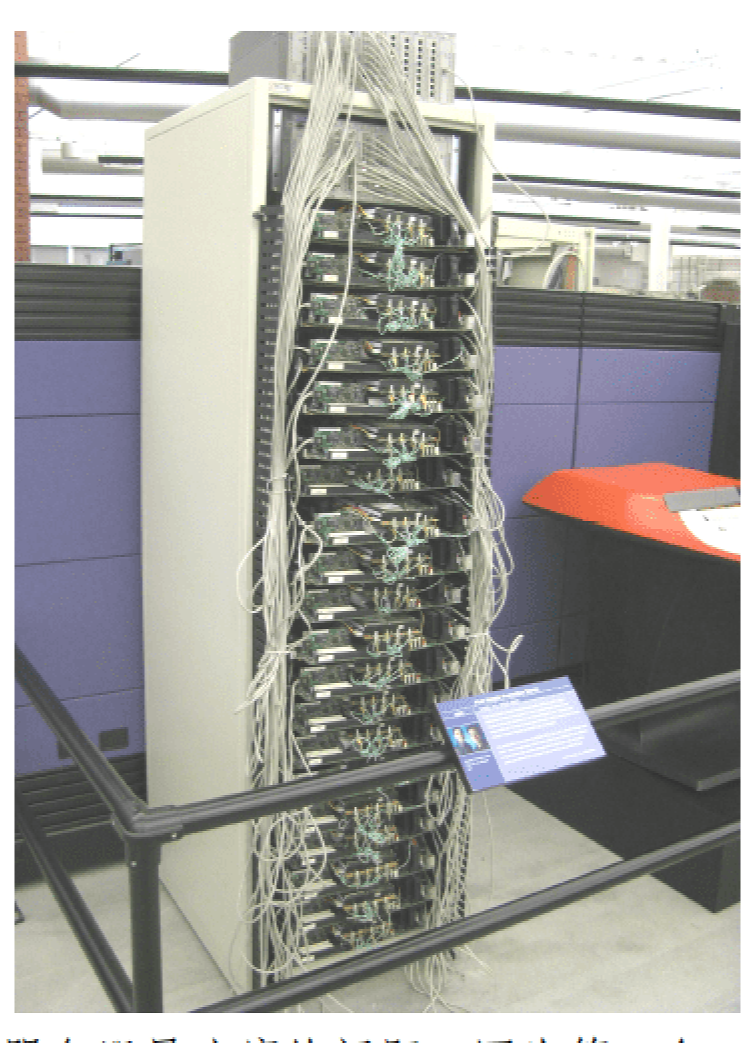
▲ 机房的成本对多服务器的企业来说将会是很大的开销

3. IT 营运成本

- 管理成本
- 折旧成本
- 设备的电费
- 服务器的租金
- 冷却费用

4. 虚拟化仍有其他支出

前面谈了这么多概念和产品，也将虚拟化的优点说明清楚，但目前全球虚拟化的比例仍然不高，虚拟机公司无所不用其极说服企业投入虚拟化的建设，但虚拟化真的能带给企业更



▲ 散热是服务器最头痛的问题，图为第一台 Google Server

高的成本效益，并且大量降低计算机化的支出吗？

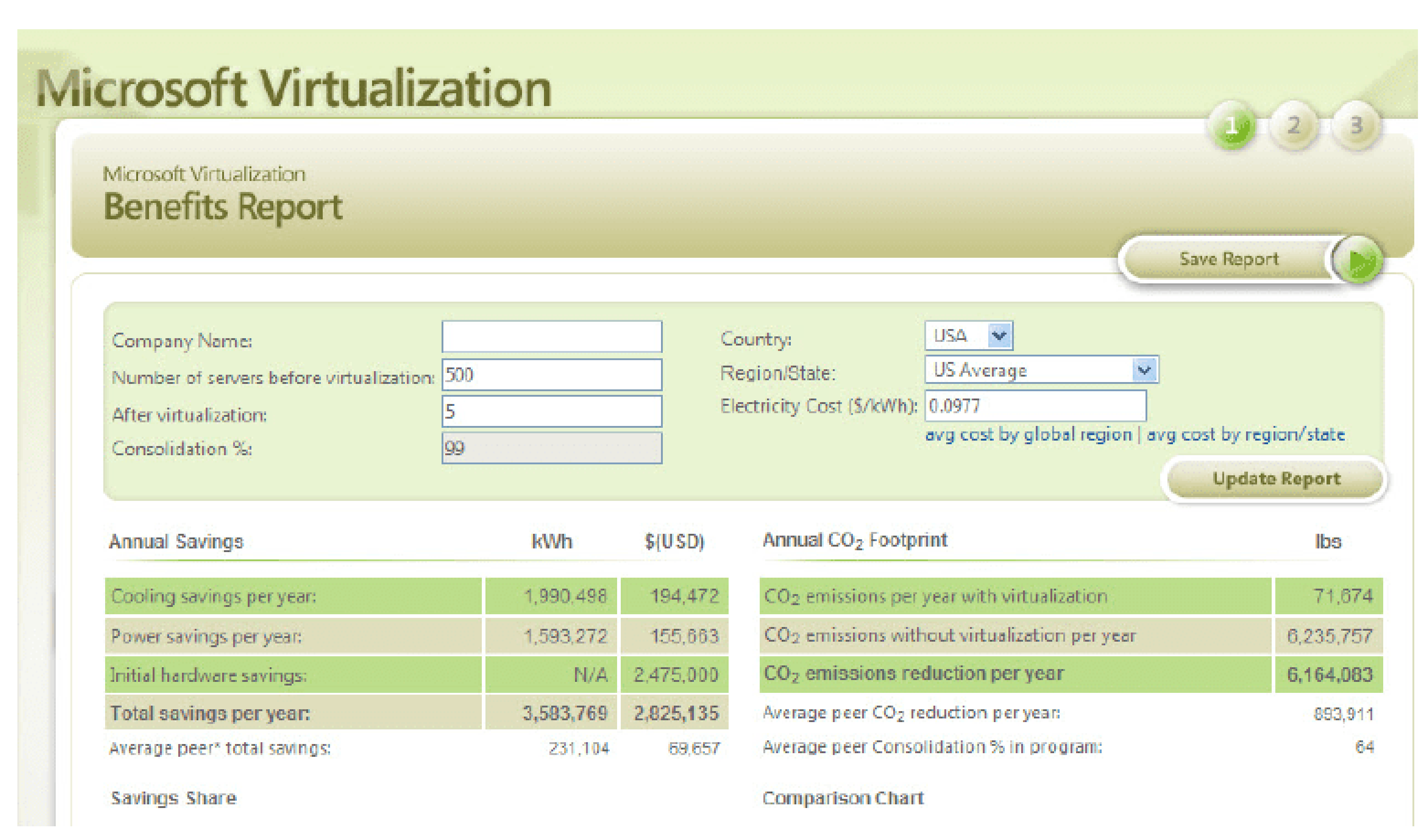
事实上企业在虚拟化时，一开始还是需要投入一些成本的，如存储设备、虚拟化软件的授权、IT 人员的训练课程等。但是以长期来看，在公司的初期资本投资、管理维护费用、人事成本上，虚拟化在长时间会产品极高的回收。不过口说无凭，在这一小节，我们就使用许多第三方公司使用的虚拟化成本计算工具，来看看不同的环境之下，虚拟化之后的成本节省。



▲ 最显著的支出就是存储设备

7.2.2 服务器合并 TCO/ROI 计算

虚拟机产品大厂 VMware 这几年不断向企业 IT 整体结构进军，希望能将虚拟化导入企业的基本建设中，而非只是一单一产品或解决方案。当然 VMware 在推出自己的产品之前，也委托了第三方的调查机构推出了成本估算工具，让企业在进入虚拟化之前，可以先了解自己是否需要虚拟化。而借着这个工具，可以算出在虚拟化之后的成本节省以及整体回报（TCO 及 ROI）。

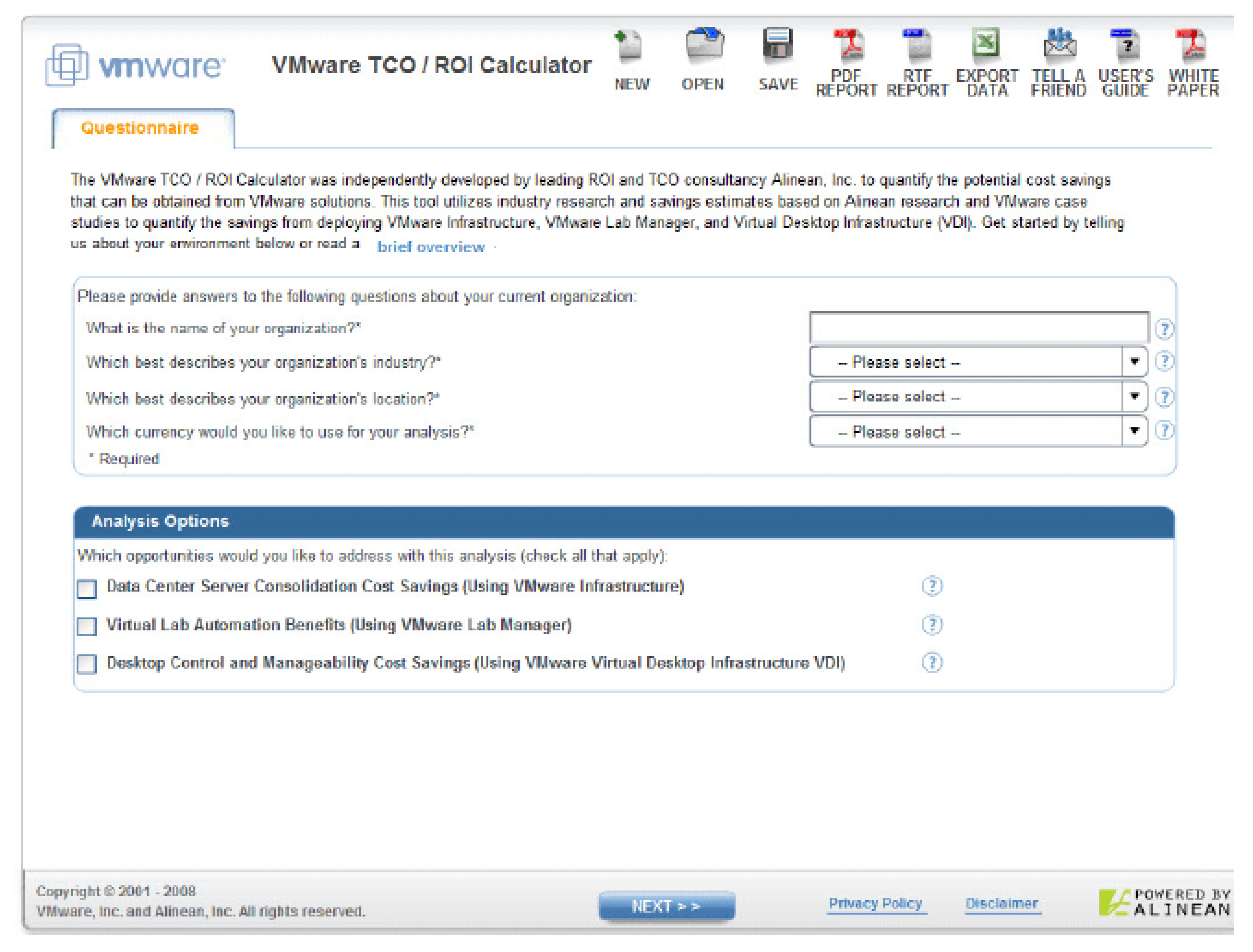


▲ 微软的 Hyper-V 也有自己的 TCO/ROI 计算器

VMware 使用一个网页的工具来计算虚拟化之后的 TCO 和 ROI，在输入一些基本的问题之后，利用其内部的计算公式来算出最符合你的情况的报告。这些报告可下载回来直接给企业业主参考，因此在真正进行虚拟化之前是很有参考价值的。我们就来看看这个工具的使用方法。

1. 了解 VMware 的 ROI 工具

进入 http://www.vmware.com/products/vi/calculator.html 这个网页，显示的就是标准 VMware TCO/ROI 计算器。基本的操作就是填写问卷，也可以将报告存成 PDF、RTF、EXCEL 下载或是存储，当然也可以在线阅读。下面就是一个最基本的例子。



▲　在线的 TCO/ROI 计算工具

举例来说，某公司机房服务器目前的硬件配置如下：

2. 未虚拟化之前的硬件配置

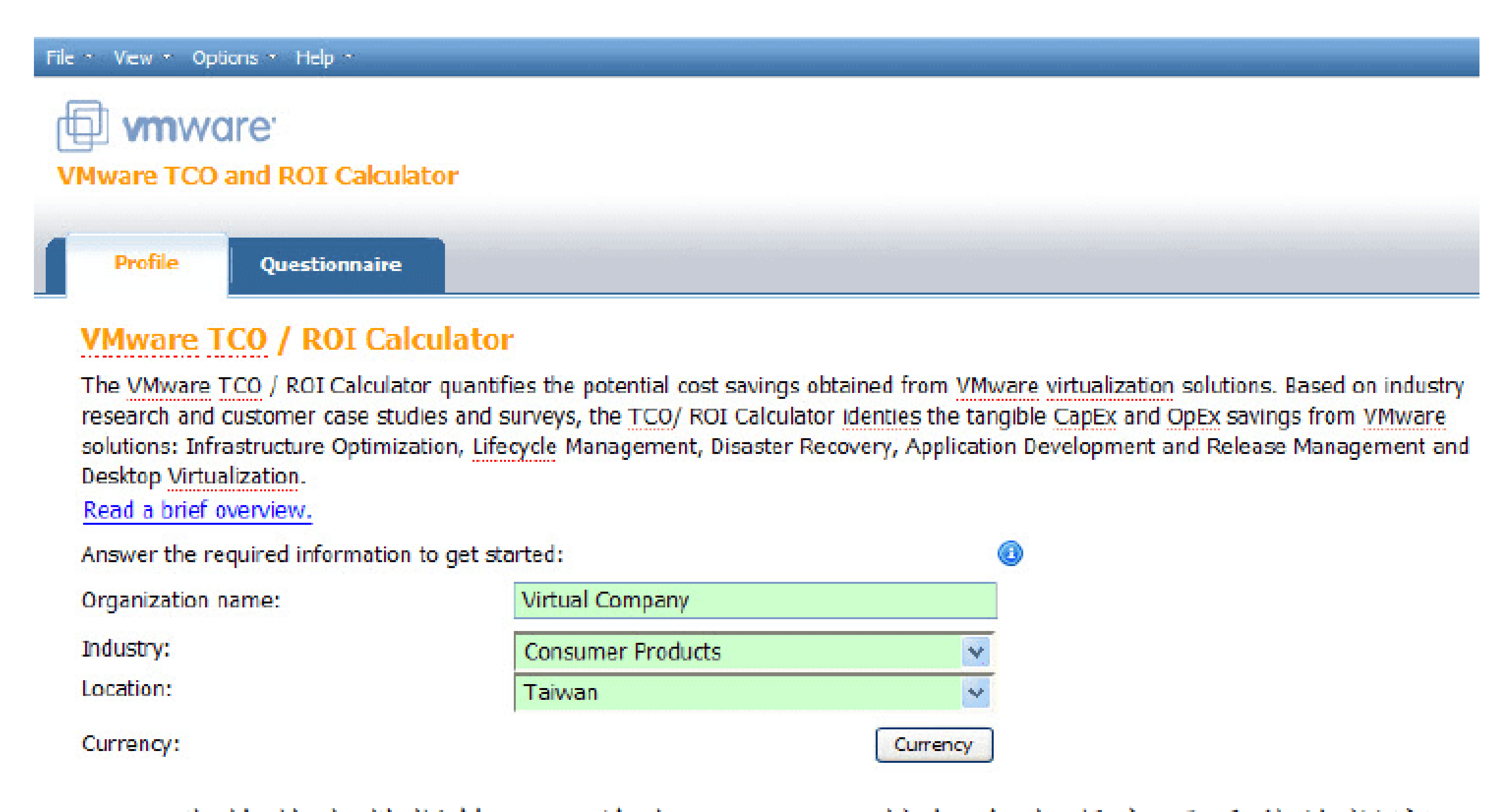
- 单 CPU 的服务器：100 台
- 双 CPU 的服务器：200 台
- 四 CPU 的服务器：50 台
- 购买新服务器评估的人工：20 人 1 小时
- 目前存储设备总容量：350 台服务器×100GB（本机硬盘）+SAN（5 000G）=40 000G。
- 连入 SAN 设备的机器百分比：20%

那么虚拟化三年后，能省下多少钱呢？接下来就用 VMware 提供地 TCO/ROI 工具来精算。

3. 使用 VMware TCO/ROI 计算器的基本数据

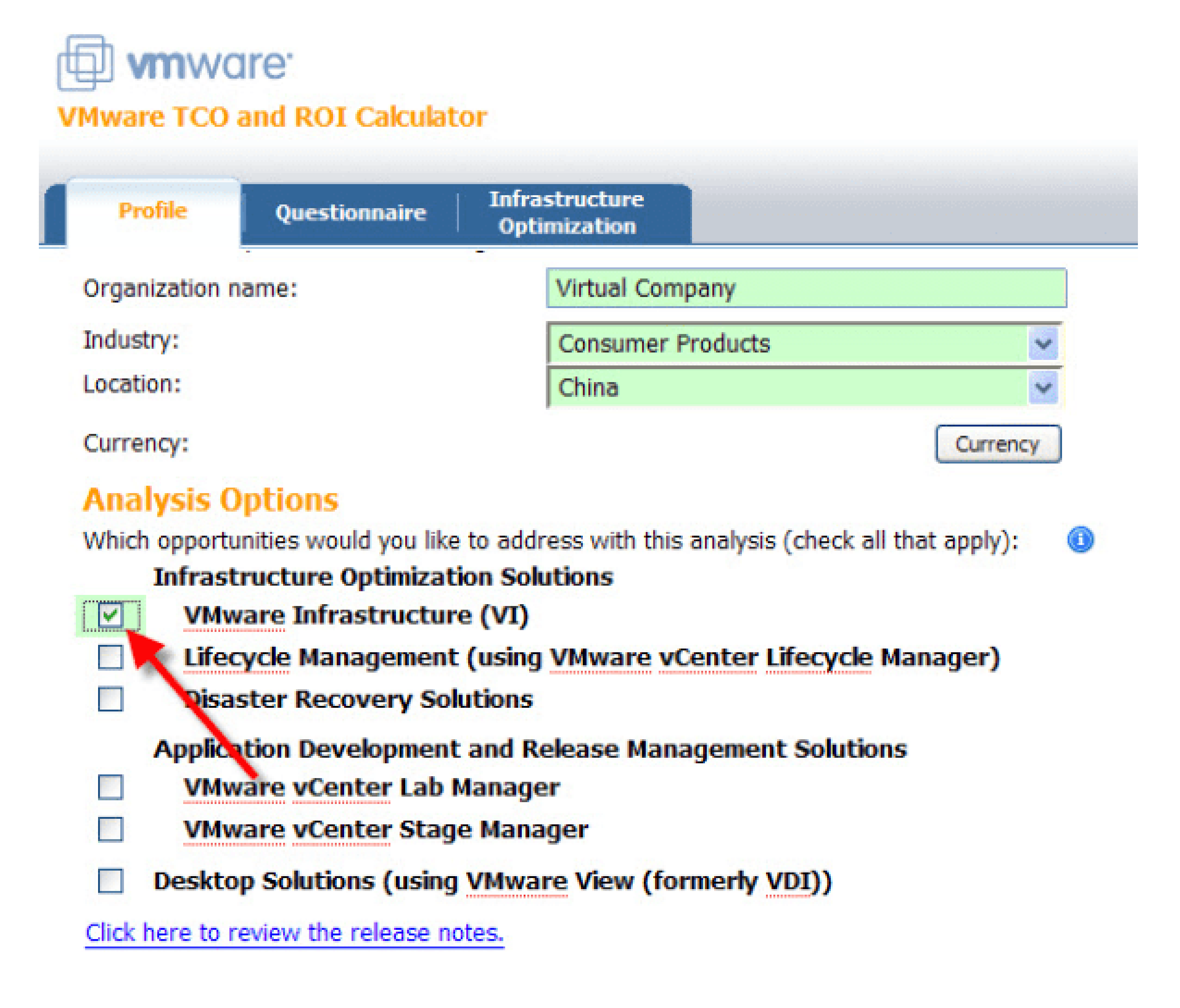
首先我们输入公司名称，在正式的报告上，你需要一个公司的名称。

接下来选择你的行业、地点等。在选择完地点之后，系统会很细心地将这个地点的币值也选择出来。



▲　先将基本数据输入，单击 Currency 按钮会出现变更币种的视窗

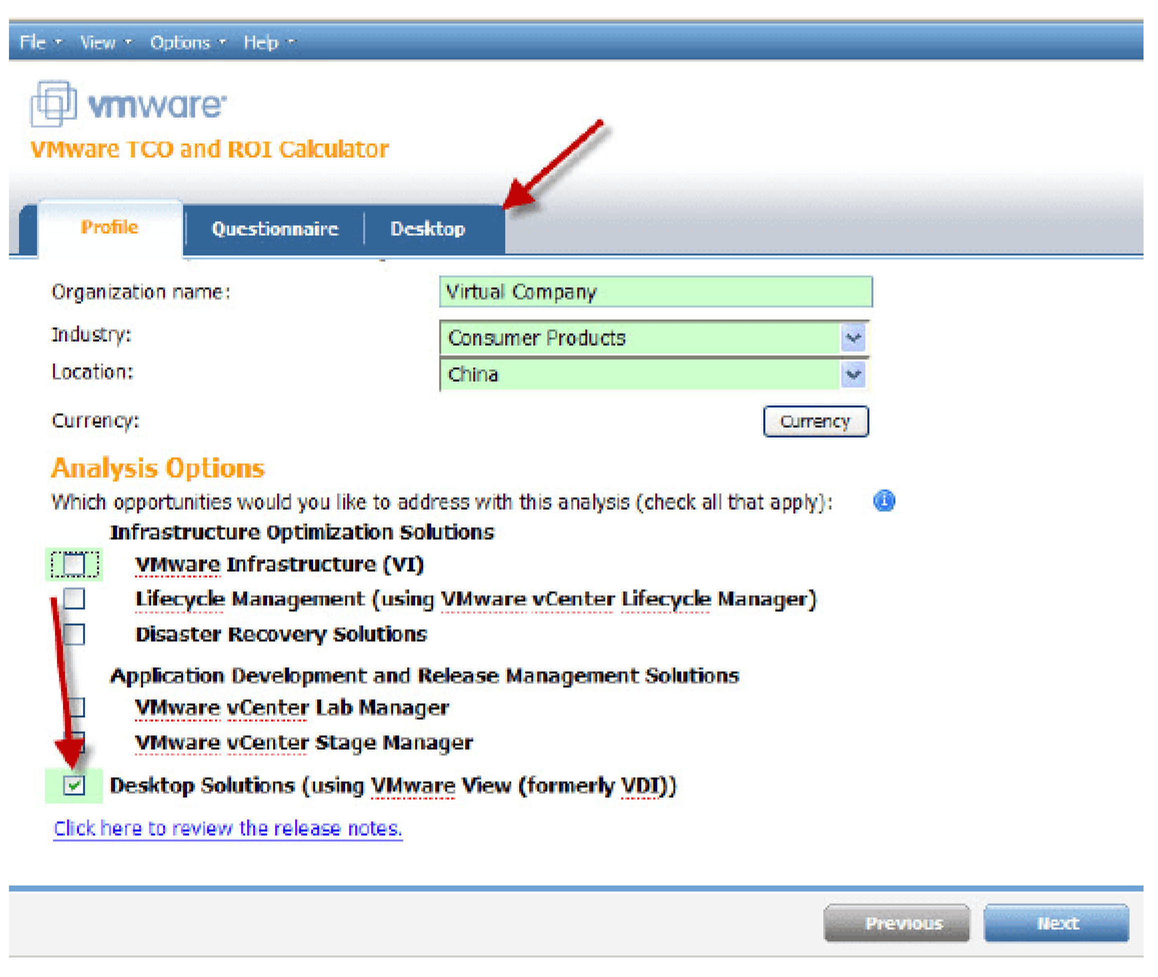
当选择完基本数据后，我们需要选择想要虚拟化的领域。一共有三个选项。如果你所属的企业环境有很多服务器，你希望能将服务器虚拟化，那么就选择第一项。



▲ 选择 VMware VI 这一项

如果你需要的是一个实验的环境，并且完成一些自动测试（软件或流程），那么可以选择 Virtual Lab。

如果你的企业服务器不多，但每一个人都有一个桌上计算机或手提计算机，那么你需要的就是桌面集成，选择 Desktop，当然以上的选择可以复选。



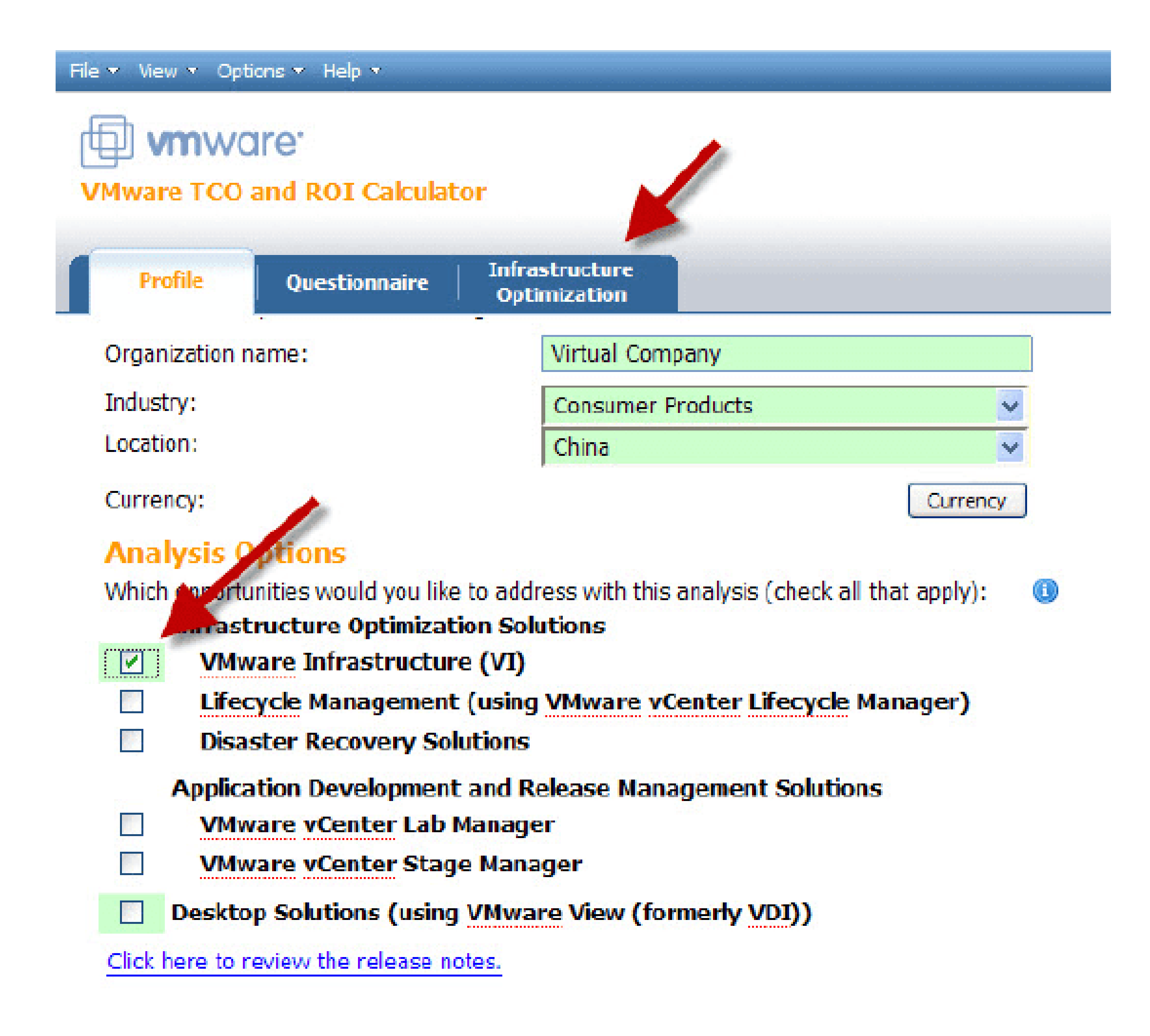
▲ 桌面虚拟化的选项

4. 服务器集成的 ROI/TCO 计算

在了解了基本资料之后，我们就可以根据需求来进行 ROI 的计算。在服务器集成部分，主要就是将多台实体服务器上的多个服务转移到虚拟机上。以现代的硬件实力来说，平均的合并率可以高达 1：20，换句话说，可以在一台功能较强的服务器上运行 20 个虚拟机，当然省下的钱不一定能到 1：20，但整个成本也有明显的下降，我们就来看看有关服务器集成的 ROI/TCO 运算。

5. 服务器集成部分

首先我们进入该网页，并且填写好所属行业及地点、币值。选择 VMware Infrastructure（VI）这一项。



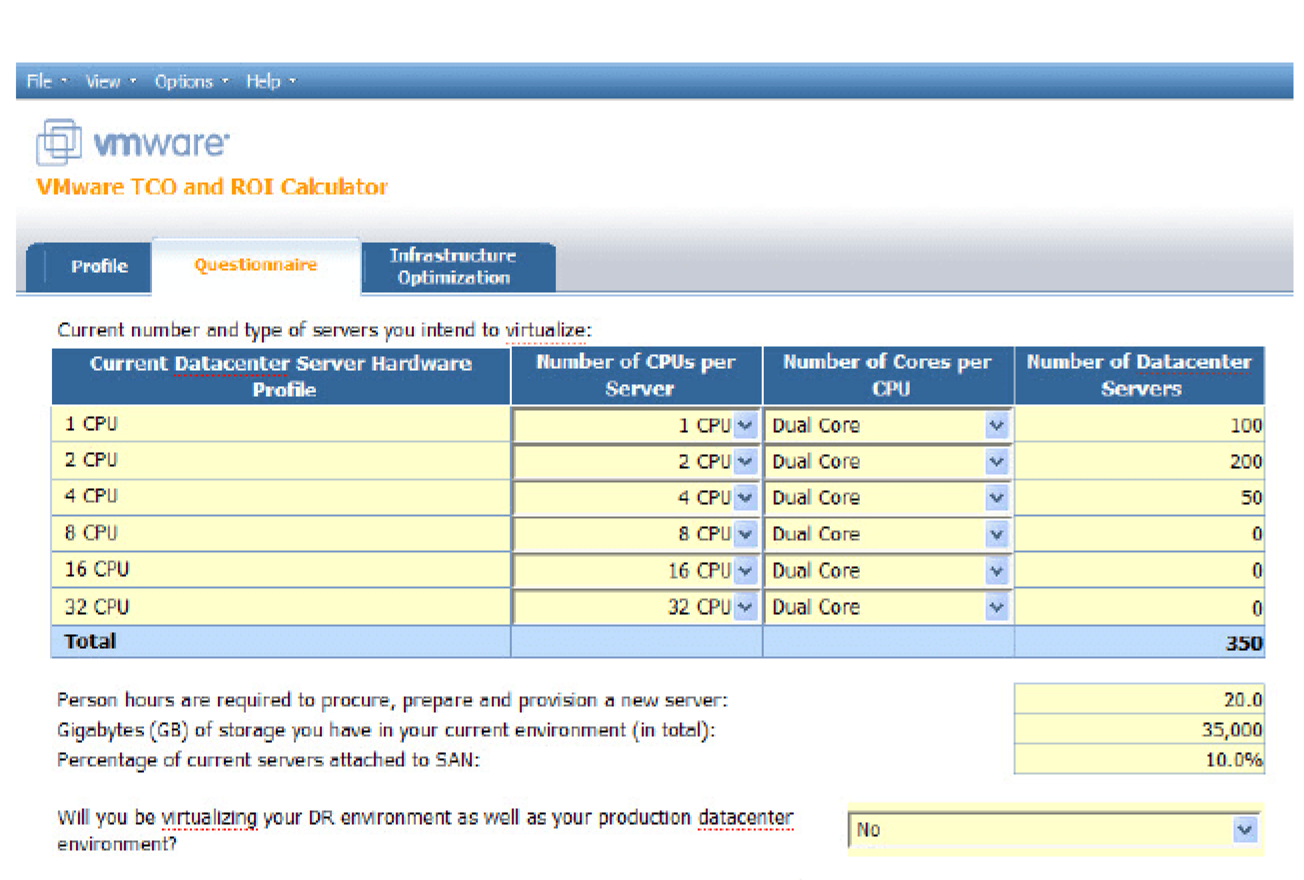
▲ 选择选项

然后我们在第一个画面，选择目前没虚拟化之前的主机数量。这是以 CPU 为主来计算，举例来说，目前的 Data Center 如果有 100 台 1 个 CPU 的主机、200 台 2 个 CPU 的主机、50 台 4 个 CPU 的主机，就照着填入，注意这些都是以双核的 CPU 为默认值。

然后的问题是有关管理成本，在购买、准备以及分配硬件资源时，需要多少人工（单位为 man hour，如 5 个人，每个人要花 4 小时，就填入 20）。

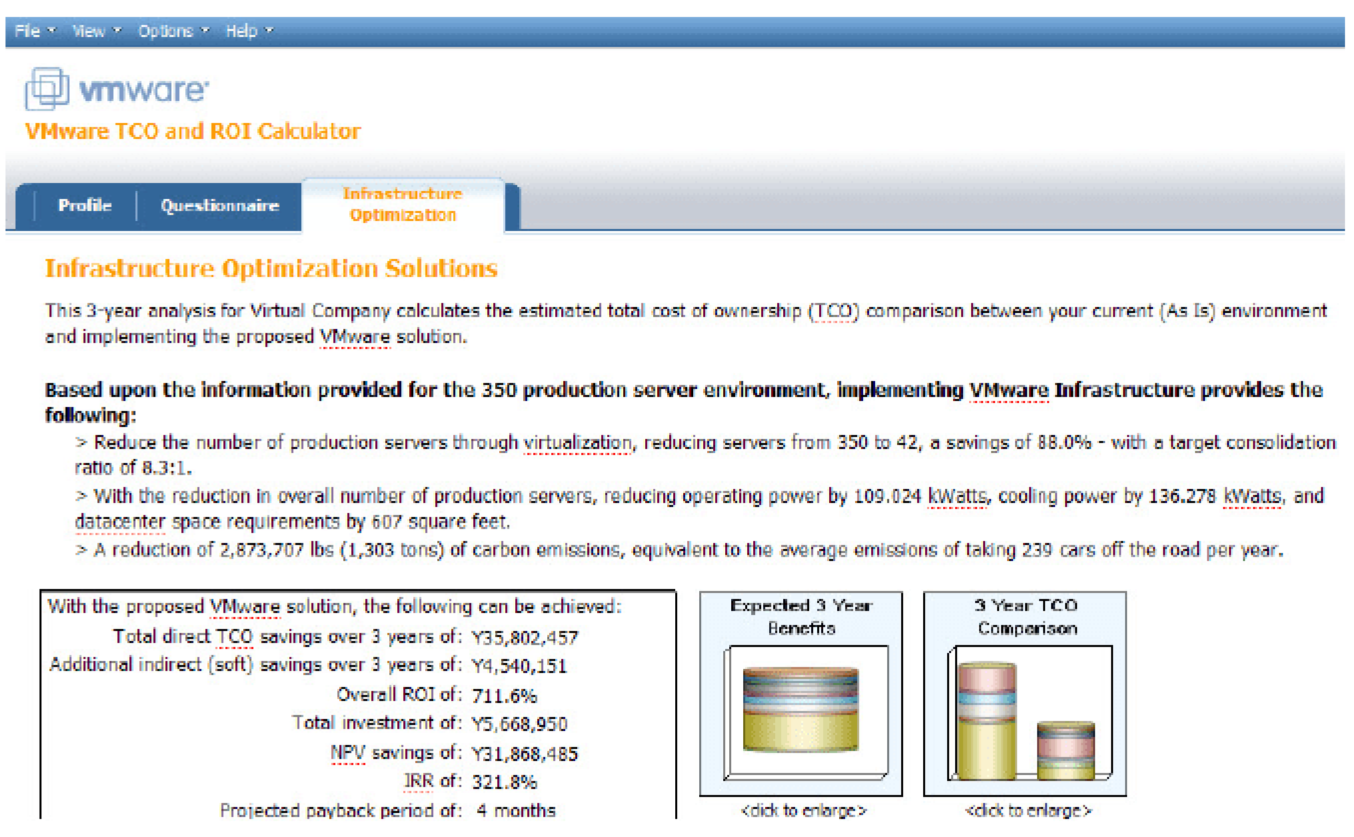
接下来是目前所有的服务器的存储空间总容量。这里包括了单机的硬盘以及网络存储，如 SAN、NAS 或是 iSCSI 等。举例来说，我们如果有 350 台服务器，每台都有 100G 的硬盘如果有做成 RAID，就只有 30000 G，再加上一个 5 000G 的 SAN，就填入 30000+5 000=35 000G。

最后一个问题就是目前连入 SAN 的机器有多大的百分比，如果服务器有 20%连入了 SAN，就填入 20%。填完之后单击 Next 按钮，当然在下方你可以更改细节的部分，我们暂时使用缺省值。



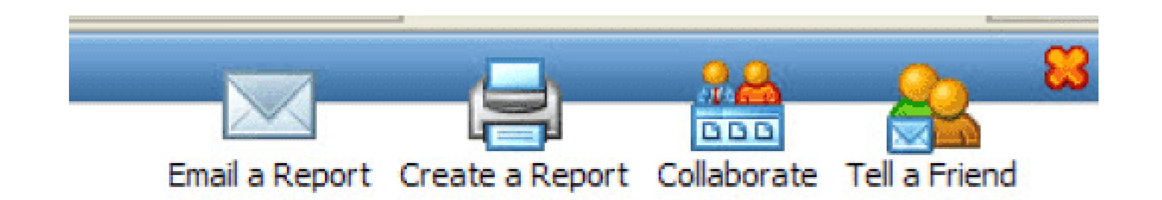
▲ 将所有数据填入

填完之后单击下图所示 Next 按钮进入下一页。进入下一页之后，你会发现这个 TCO/ROI 电子表格会列出每一单项的数据，并且将没有虚拟化和虚拟化之后的每一个细项作比较。



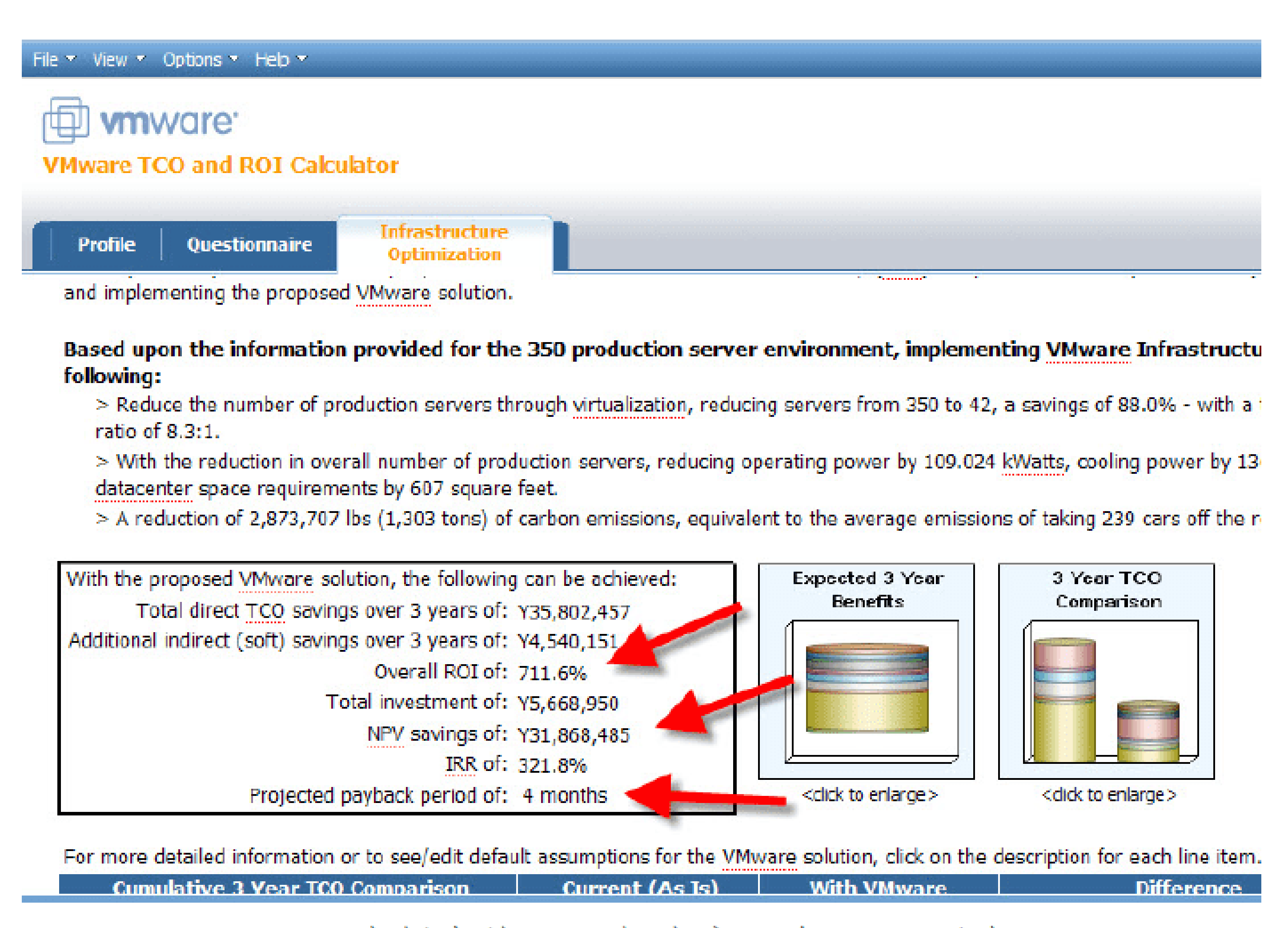
▲ 报表计算出来，有许多细项

你可以直接单击下图所示图表显示详细的比较图，或是单击上面按钮选择输入至文件或是电子邮件等功能。



▲ 可以下载不同格式的报表

在下图显示的最上面的黑体字，显示的是 ROI 的回报率是 711.6%，相当高的数字！NPV Savings 为 3 年省下的花费净值（以硬件 9.5%的折旧率）。第三个则是回本的时间，本例中是 4 个月。



▲ 本例中的 ROI 相当高，有 711%以上

6. TCO/ROI 计算器详细说明

这个 TCO/ROI 计算器的表格中的项目还是很详细的，在 VMware 的 TCO/ROI 表格中，共有

下面几个大类。

- 服务器硬件：虚拟化前后的三年硬件成本比较，是最省钱的部分。
- 存储设备：要注意的是，此项为虚拟化时必须添加的设备，因此会花钱，投资回报率为负值。
- 网络设备：主要是网卡和交换机、路由器的设备，虚拟化后多为正值。
- 冷却及电力：电力和冷却服务器的花费，虚拟化后多为正值。
- 场地租金：放服务器和设备的场地租金，虚拟化后多为正值。
- 服务器规划：这是规划服务器的人工费用，虚拟化后多为正值，因为虚拟化时，我们可以用模板来直接套用而省去安装服务器的麻烦。
- 服务器管理费用：计算机少了，再加上一些防灾的功能（VMotion，DRS，HA），这个费用也会少，虚拟化后多为正值。
- 因灾难而损失的费用：VMware 在这里认定为正值，即灾难发生的损失较少。
- 服务器停机时间损失：由于虚拟化后，停机时间较少，虚拟化后多为正值。
- VI 软件及授权费用：这是虚拟化中最花钱的部分，此例中，一台服务器安装 VI3 要 4 万，这比一台 4CPU 的服务器还要贵了，当然是负值。
- 其他软件授权费：有些软件在虚拟机上运行要授权费，这里由用户输入。
- VI 设计首部署：VI 的设计、规划及部署也是收费的，因此为负值。
- VI 教育训练：使用 VI 必须上课，当然如果你熟读本书，这个费用可以省下不少。
- 计划停机使用 VMotion 的好处：使用 VMotion 可以让计划停机的时间变少。
- 计划停机使用存储 VMotion 的好处：使用存储 VMotion 的好处。
- 使用 DRS 的动态加载好处：使用 DRS 获得的利益。
- 使用 vCenter 替服务器做补丁更新的利益：使用 vCenter 之后，就不需要一台台更新服务器的 HotFix 或是 SP 了。



VMware Infrastructure (VI) Benefits	
Datacenter Server Hardware	Y27,067,262
Datacenter Server Storage	Y844,880
Datacenter Server Networking	Y1,786,699
Datacenter Server Power and Cooling Consumption	Y6,231,112
Datacenter Server Space	Y4,929,787
Datacenter Server Provisioning	Y1,842,086
Datacenter Server Administration	Y10,247,625
Datacenter Server Unplanned Downtime (Indirect)	Y12,971,860
Investment Required	
VMware Infrastructure Software Licenses	Y0
VMware Infrastructure Support and Subscription	Y0
Additional Software Licensing Costs (if any)	Y0
VMware Infrastructure Design, Plan and Deployment Services	Y0

▲ 这是服务器合并的分类项，每一项中都有细节

7. 金额说明

每一个分类都有虚拟化之前、之后、节省的成本等。

（1）三年内回收 TCO/ROI 的结果说明

举例来说，下图中第一个大类为服务器硬件成本，没有虚拟化之前是 2700 多万，虚拟化之后是 500 多万，省了 2000 多万。

右边有一个百分比，下图中的 80%，表示这项投资的回收率，如果这个数值的负值，表示这个类别是需要有支出的。

如果不想把这个类别列入计算，将最右边的 Exclude 选中，就不会列入计算了。

For more detailed information or to see/edit default assumptions for the VMware solution, click on the description for each line item.

Cumulative 3 Year TCO Comparison	Current (As Is)	With VMware Infrastructure	Difference ($ and % savings)		Exclude?
VMware Infrastructure (VI) Benefits					
Datacenter Server Hardware	Y27,067,262	Y5,414,900	Y21,652,361	80.0%	☐
Datacenter Server Storage	Y844,880	Y2,965,326	(Y2,120,446)	-251.0%	☐
Datacenter Server Networking	Y1,786,699	Y333,109	Y1,453,590	81.4%	☐
Datacenter Server Power and Cooling Consumption	Y6,231,112	Y1,288,490	Y4,942,622	79.3%	☐

▲ 每一项都有详细的 ROI

直接到本页最下方，可以看到 Total Benefits 的总和为 3700 多万，而最下面的 Total Costs 为 560 多万，扣除原来服务器的成本，竟然可以省下 4000 多万元！

File · View · Options · Help ·

vmware

VMware TCO and ROI Calculator

Profile | Questionnaire | Infrastructure Optimization

Datacenter Server Administration	Y446,047	Y510,275	Y583,754	Y1,540,076
Datacenter Server Unplanned Downtime (Indirect)	Y1,513,384	Y1,513,384	Y1,513,384	Y4,540,151
Total Benefits	**Y11,521,421**	**Y12,621,050**	**Y13,844,640**	**Y37,987,111**

Additional Benefits from VI	Year 1	Year 2	Year 3	Total
VMware VMotion Cost Savings for Server Planned Maintenance	Y165,861	Y189,080	Y215,553	Y570,494
VMware Storage VMotion Cost Savings for Storage Planned Maintenance	Y395,353	Y450,700	Y513,795	Y1,359,849
VMware DRS Cost Savings for Dynamic Load Balancing	Y778,998	Y856,898	Y942,588	Y2,578,483
VMware vCenter Update Manager Cost Savings for Guest/Host Patching	Y1,022,100	Y1,165,196	Y1,328,325	Y3,515,621
Total Additional Benefits	**Y2,362,313**	**Y2,661,874**	**Y3,000,261**	**Y8,024,448**

Expected Investment in VMware	Initial / Year 1	Year 2	Year 3	Total
VMware Infrastructure Software Licenses	Y1,927,008	Y192,704	Y211,970	Y2,331,682
VMware Infrastructure Support and Subscription	Y404,674	Y445,141	Y489,650	Y1,339,465
Additional Software Licensing Costs (if any)	Y0	Y0	Y0	Y0
VMware Infrastructure Design, Plan and Deployment Services	Y1,864,688	Y0	Y0	Y1,864,688
VMware Infrastructure Training	Y133,116	Y0	Y0	Y133,116
Total Costs	**Y4,329,485**	**Y637,845**	**Y701,620**	**Y5,668,950**

Return on Investment (ROI)	**711.6%**
NPV Savings (3 year, discount rate = 11.0%)	**Y31,868,485**

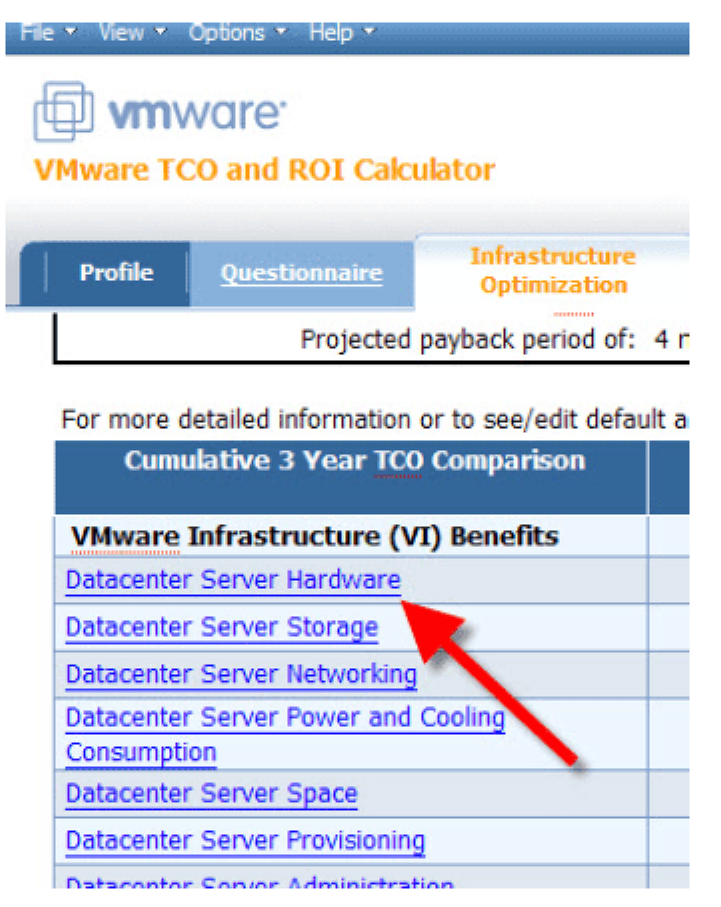
▲ 省下的钱和成本比较起来，虚拟化是很划得来的

（2）分类详细计算

在这个结果中，每一个物品的单价是从何而来的呢？这些数据是如何计算出来的？在每一个分类中，你可以进入查看细节，我们就以服务器硬件成本为例。

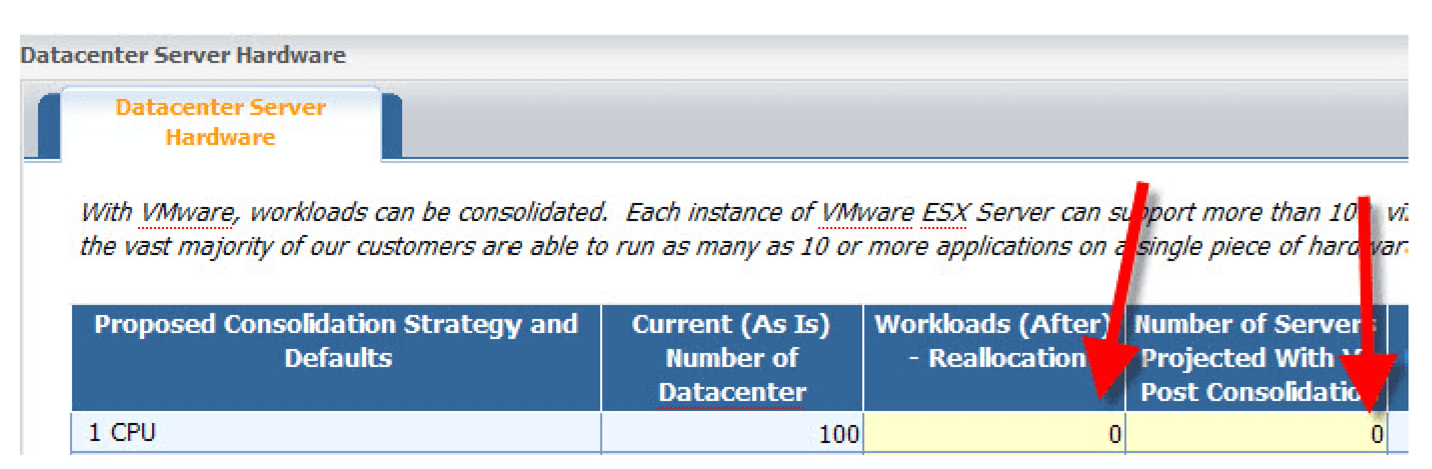
（3）服务器硬件成本价钱精算

我们直接选择 Data Center Server Hardware 选项，进入查看细目。



▲ 查看详细说明

首先我们发现单 CPU 的 100 台服务器在虚拟化之后全部没有了，因为单 CPU 是最不合经济效益的服务器。



Proposed Consolidation Strategy and Defaults	Current (As Is) Number of Datacenter	Workloads (After) - Reallocation	Number of Servers Projected With VI Post Consolidation
1 CPU	100	0	0

▲ 单 CPU 的服务器全部取消，因为其性价比太差

原来 350 台服务器中，双 CPU 的机器有 200 台，但在这里，将原来单 CPU 的 100 台先并过来，并且再将这 300 台实体服务器通过服务器集成减少到 38 台。

Proposed Consolidation Strategy and Defaults	Current (As Is) Number of Datacenter	Workloads (After) - Reallocation	Number of Servers Projected With VI Post Consolidation
1 CPU	100	0	0
2 CPU	200	300	38
4 CPU	50	50	4
Total	350	350	42

▲ 将 100 台单 CPU 的服务并到双 CPU 上，再虚拟化之后变成 38 台

原来 50 台 4CPU 的服务器也被合并成 4 台。虚拟化之后，实体服务器的数量从 350 台变成 42 台，合并率为 8.3 比 1。

Proposed Consolidation Strategy and Defaults	Current (As Is) Number of Datacenter	Workloads (After) - Reallocation	Number of Servers Projected With VI Post Consolidation	Number of CPUs Projected (After) VI
1 CPU	100	0	0	0
2 CPU	200	300	38	76
4 CPU	50	50	4	16
Total	350	350	42	92

Proposed Target Consolidation Ratio 8.3 to 1

Server Hardware	Current (As Is)	Proposed Savings with VI	With VI (Projected)
1 CPU			
Number of servers	100		0
Average amortized cost per server per year	Y13,267		Y6,637
Total	Y1,326,699	Y1,326,699	Y0
2 CPU			
Number of servers	200		38

Close

▲ 合并率为 8.3 比 1

接下来是服务器的单价，在虚拟化之前，单 CPU 的服务器在下图中是 6 万多元，在虚拟化之后，由于不使用单 CPU 的服务器，因此数量是 0，总成本也是 0。

Server Hardware	Current (As Is)	Proposed Savings with VI	With VI (Projected)
1 CPU			
Number of servers	100		0
Average amortized cost per server per year	¥13,267		¥6,637
Total	¥1,326,699	¥1,326,699	¥0
2 CPU			
Number of servers	200		38
Average amortized cost per server per year	¥21,565		¥34,502
Total	¥4,312,972	¥3,001,881	¥1,311,091

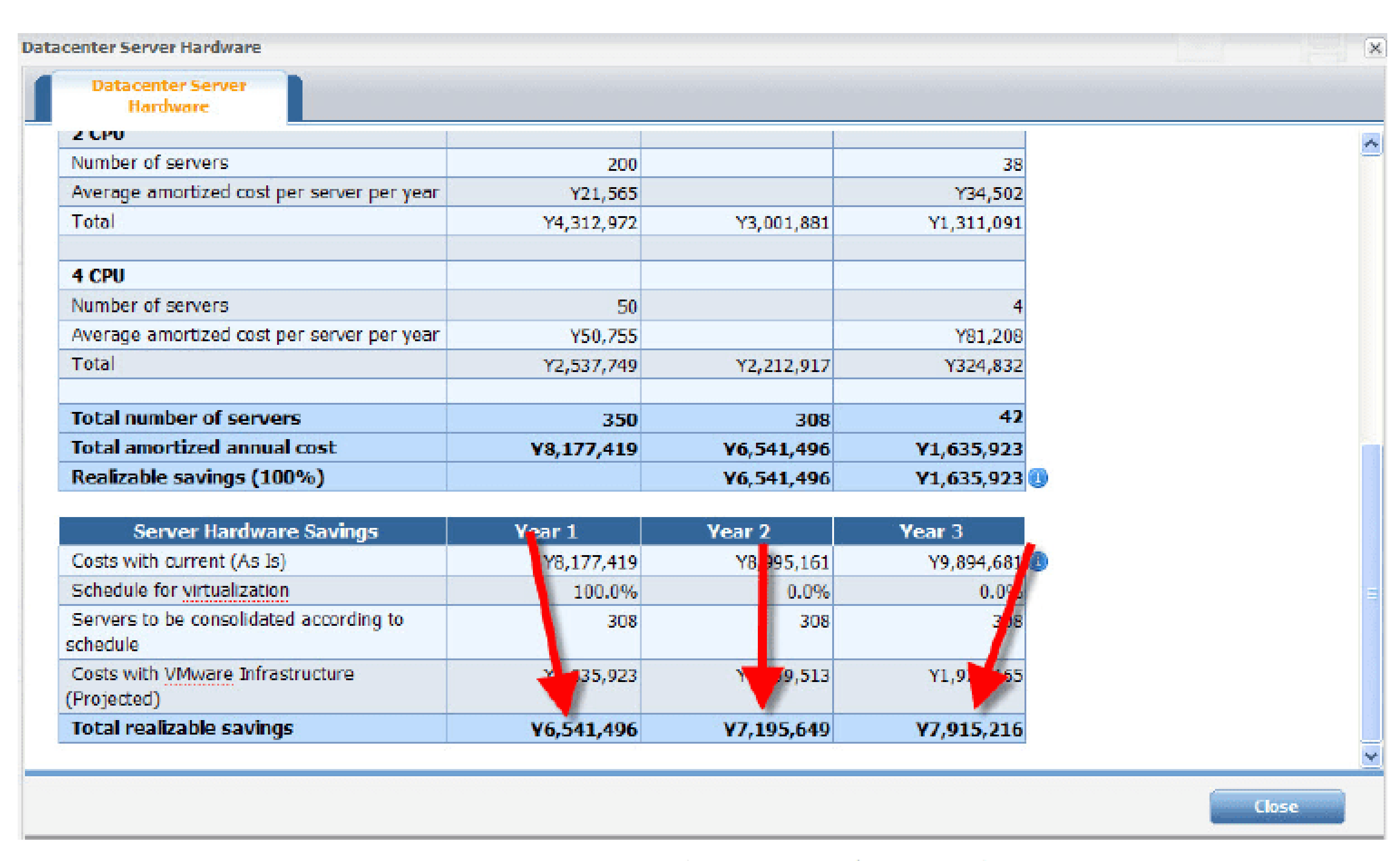
▲ 虚拟化之后的实体机的价钱变贵了，主要是添加了 CPU 和内存

双 CPU 的服务器原来的单价是 21000 元，有 200 台，但虚拟化之后需要更新内存以及其他设备，因此成本涨到 34000 元，但数量从 200 台变成 38 台，因此双 CPU 的硬件总成本 430 万降为 131 万。

同理 4CPU 的服务器总成本也从 250 多万降成 32 万。

因此光在硬件服务器这一块，虚拟化的总成本每年可以省 600 多万到 790 万左右。

单击 Return To Calculator 可以回到报表中。


Datacenter Server Hardware

2 CPU			
Number of servers	200		38
Average amortized cost per server per year	¥21,565		¥34,502
Total	¥4,312,972	¥3,001,881	¥1,311,091
4 CPU			
Number of servers	50		4
Average amortized cost per server per year	¥50,755		¥81,208
Total	¥2,537,749	¥2,212,917	¥324,832
Total number of servers	**350**	**308**	**42**
Total amortized annual cost	**¥8,177,419**	**¥6,541,496**	**¥1,635,923**
Realizable savings (100%)		**¥6,541,496**	**¥1,635,923**

Server Hardware Savings	Year 1	Year 2	Year 3
Costs with current (As Is)	¥8,177,419	¥8,[illegible]95,161	¥9,894,681
Schedule for virtualization	100.0%	0.0%	0.0[illegible]
Servers to be consolidated according to schedule	308	308	[illegible]
Costs with VMware Infrastructure (Projected)	[illegible]35,923	[illegible]9,513	¥1,9[illegible]65
Total realizable savings	**¥6,541,496**	**¥7,195,649**	**¥7,915,216**

▲ 服务器硬件的前后成本差异的确很大

7.2.3 虚拟桌面管理成本计算

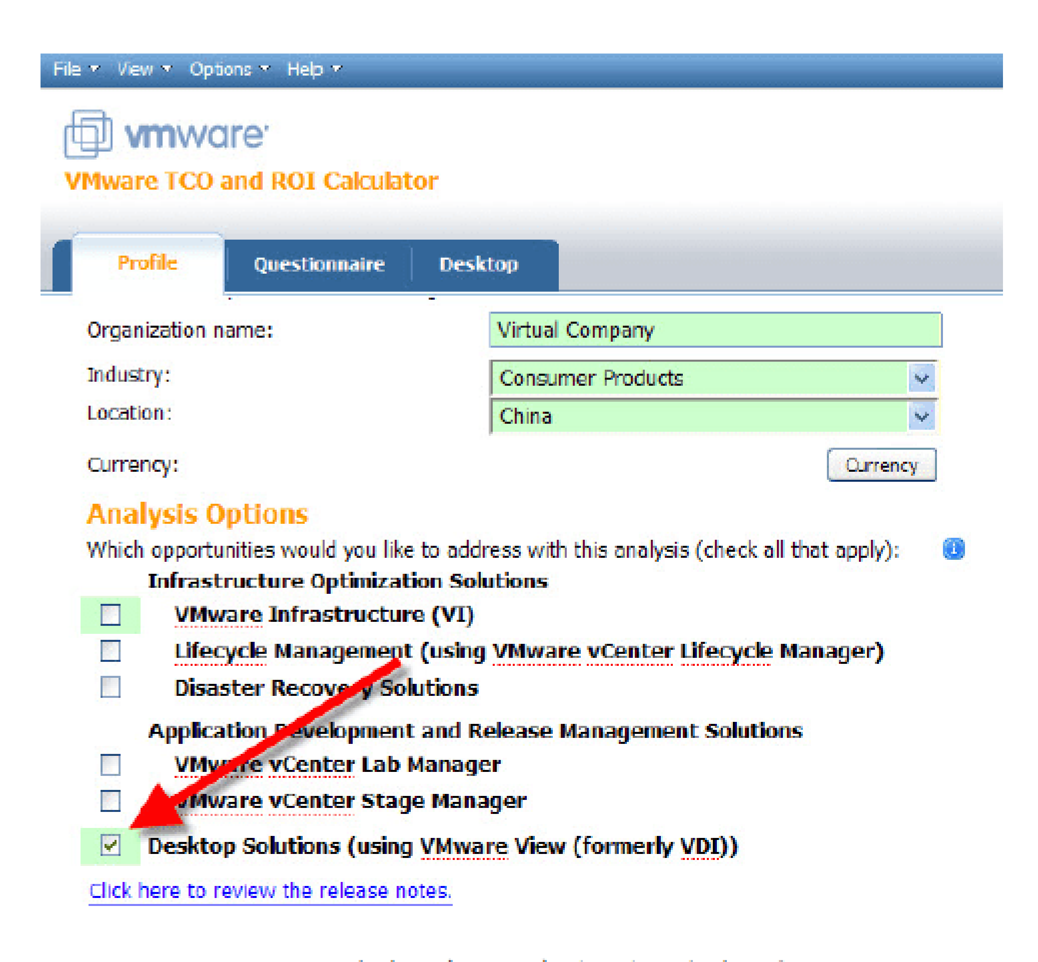
在了解了服务器合并的成本计算后，桌面虚拟化的 TCO/ROI 就相对简单多了。在桌面虚拟化的过程中，将桌面系统合并至服务器上，必须购买新的服务器、客户端设备及存储设备，再加上虚拟化的软件，因此回收率会比服务器合并较低。

1. 决定需要合并的个人计算机数量

桌面虚拟化的合并率可比服务器高一些，下面仍然是以 VMware 的 TCO/ROI 工具来计算。

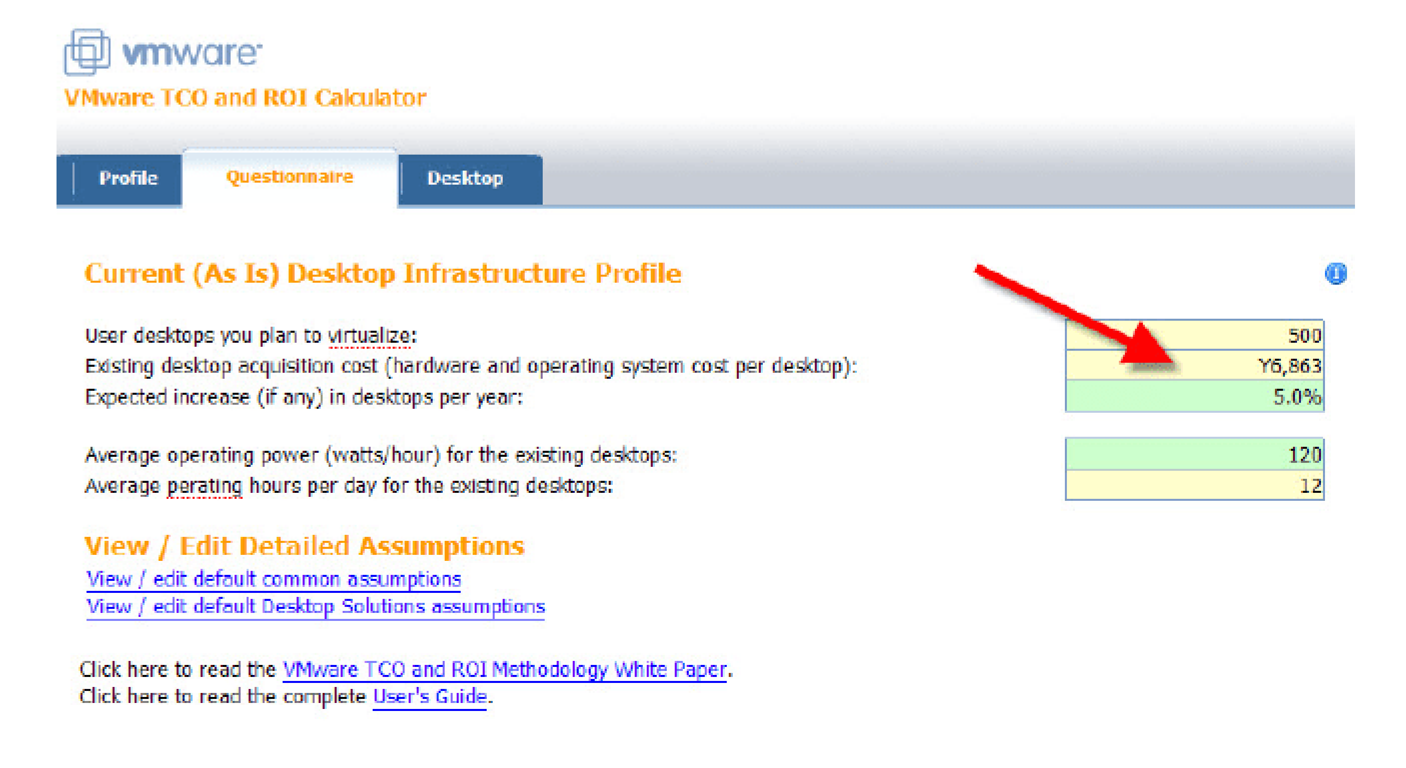
▶ 桌面虚拟化的 TCO/ROI 计算

首先仍然进入 VMware 的 TCO/ROI 计算页，此时选择 Desktop 选项。



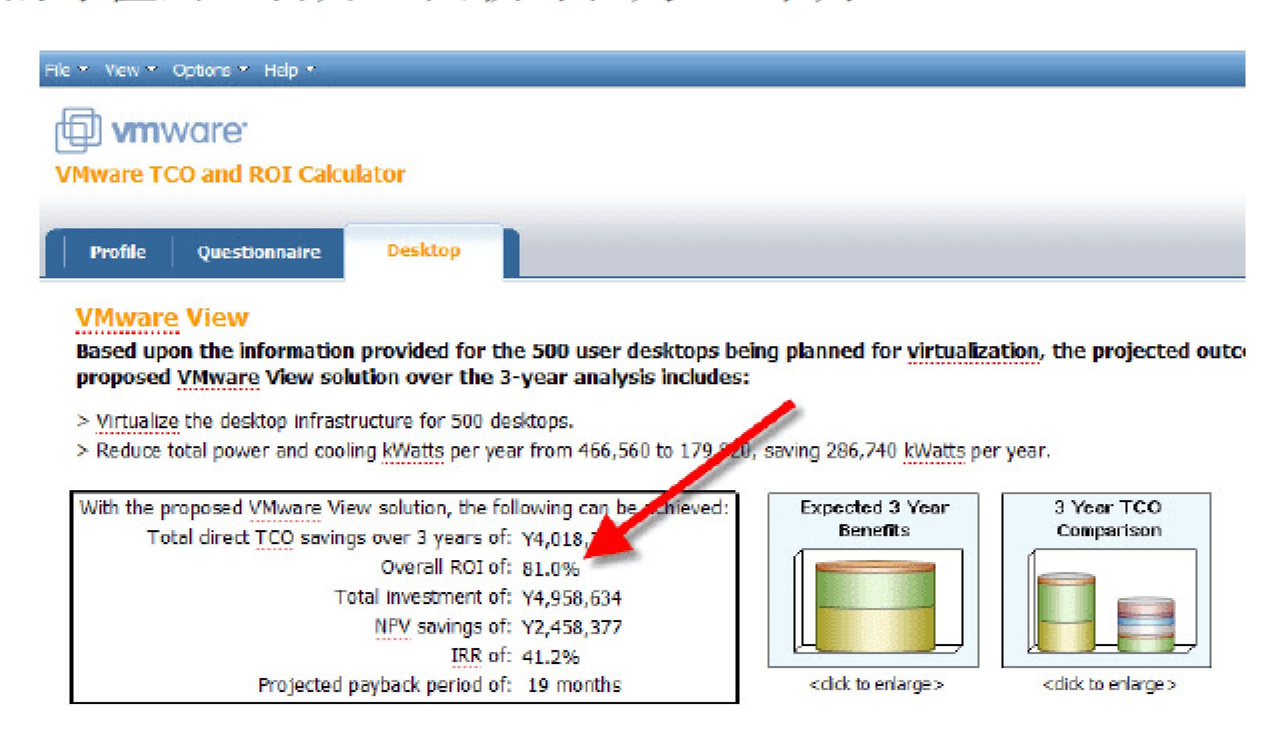
▲ 选择桌面虚拟化的部分

之后会出现如下图所示的画面，分别为需要虚拟化的桌面系统数量及每一台计算机的成本，包括了硬件成本和操作系统的成本、预估公司每年会添加多少百分比的桌面计算机以及每台计算机的电力消耗瓦数、每天开机时间等。我们就依次输入。



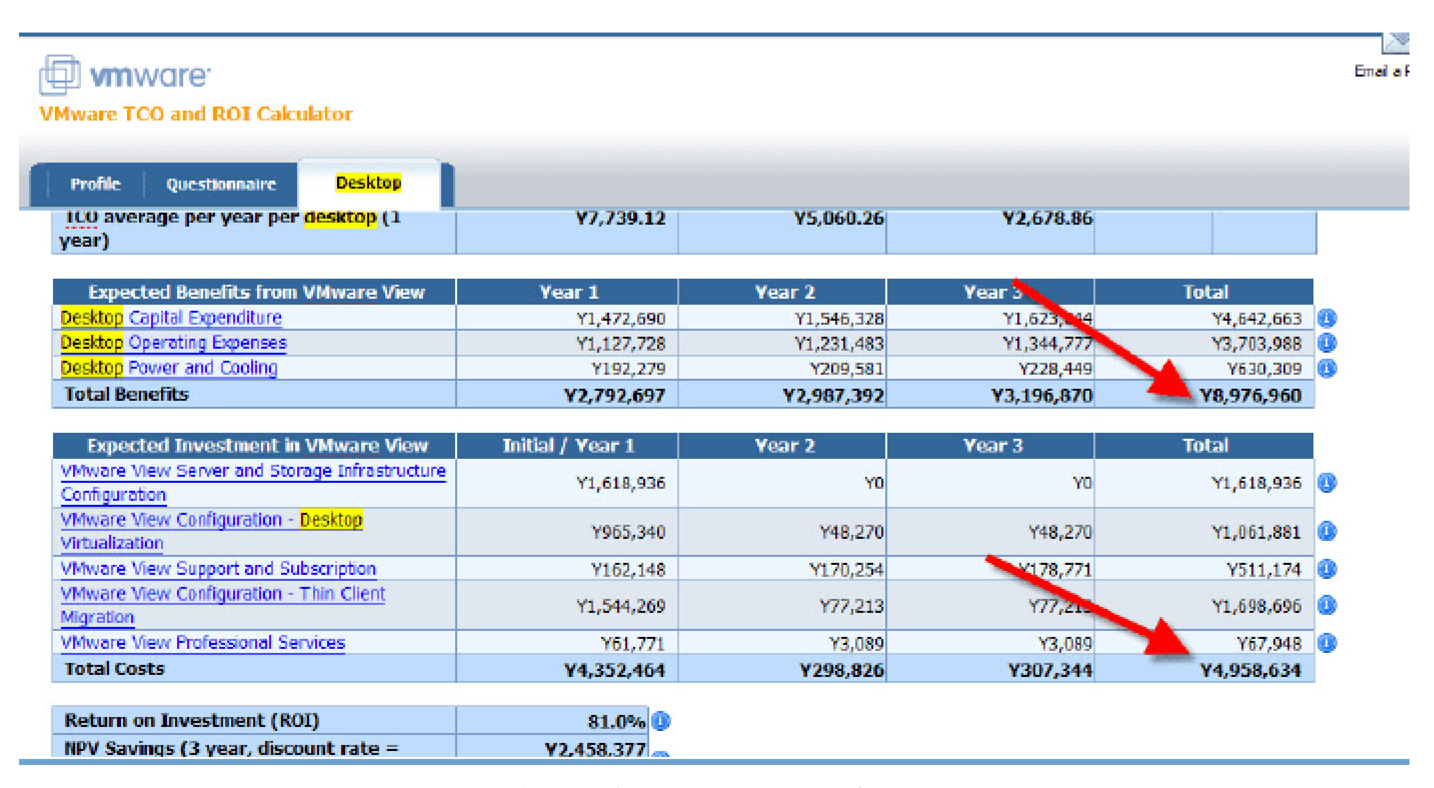
▲ 依次输入数据

输入完毕之后单击 Next 按钮，就会出现报表（如下图）。可以看到 ROI 是 80%左右，比服务器低了不少，3 年的净值约 4 百万，回收时长为 19 个月。



▲ 桌面虚拟化的 ROI 较低，但和实体比起来也有 80%以上

在表格最下面一行，每一台计算机的每年成本从从 7700 降到 5000，省了接近 2200 元，总获益为 890 万元，总支出为 490 万元。



TCO average per year per desktop (1 year)	Y7,739.12	Y5,060.26	Y2,678.86		

Expected Benefits from VMware View	Year 1	Year 2	Year 3	Total
Desktop Capital Expenditure	Y1,472,690	Y1,546,328	Y1,623,[illegible]	Y4,642,663
Desktop Operating Expenses	Y1,127,728	Y1,231,483	Y1,344,777	Y3,703,988
Desktop Power and Cooling	Y192,279	Y209,581	Y228,449	Y630,309
Total Benefits	Y2,792,697	Y2,987,392	Y3,196,870	Y8,976,960

Expected Investment in VMware View	Initial / Year 1	Year 2	Year 3	Total
VMware View Server and Storage Infrastructure Configuration	Y1,618,936	Y0	Y0	Y1,618,936
VMware View Configuration - Desktop Virtualization	Y965,340	Y48,270	Y48,270	Y1,061,881
VMware View Support and Subscription	Y162,148	Y170,254	Y178,771	Y511,174
VMware View Configuration - Thin Client Migration	Y1,544,269	Y77,213	Y77,21[illegible]	Y1,698,696
VMware View Professional Services	Y61,771	Y3,089	Y3,089	Y67,948
Total Costs	Y4,352,464	Y298,826	Y307,344	Y4,958,634

Return on Investment (ROI)	81.0%
NPV Savings (3 year, discount rate =	Y2,458,377

▲ 桌面虚拟化的回收也很高

2. 最花钱的部分

在桌面虚拟化架构上，最花钱的部分从图中可以看到是投资使用 VMware View 和一些相关硬件及配置的地方。

Investment Required				
VMware View Server and Storage Infrastructure Configuration	Y0	Y1,618,936	(Y1,618,936)	0.0%
VMware View Configuration - Desktop Virtualization	Y0	Y1,061,881	(Y1,061,881)	0.0%
VMware View Support and Subscription	Y0	Y511,174	(Y511,174)	0.0%
VMware View Configuration - Thin Client Migration	Y0	Y1,698,696	(Y1,698,696)	0.0%
VMware View Professional Services	Y0	Y67,948	(Y67,948)	0.0%
Total TCO (3 year)	**Y11,608,669**	**Y7,590,343**	**Y4,018,325**	**34.6%**
TCO average per year per desktop (1 year)	**Y7,739.12**	**Y5,060.26**	**Y2,678.86**	

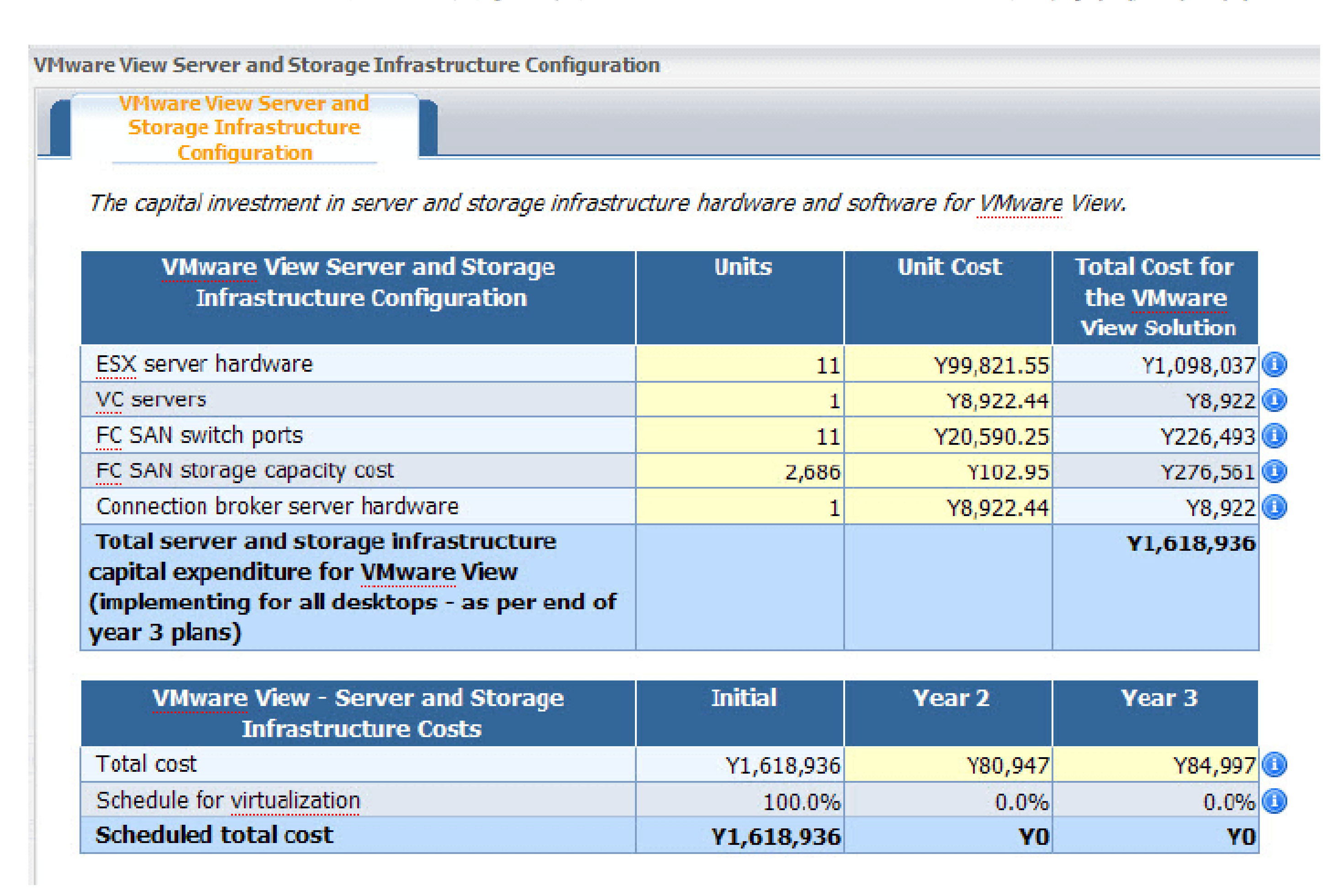
▲ 这是花费最多的地方

3. 桌面虚拟化的花费

首先是 VMware View Servers，每台需要将近 10 万元，使用的服务器都是 4 CPU。

VC Server 是一台运行 Windows Server 2003 的数据库，也需要购买：

- FC SAN 交换机：这是在桌面化时需要的。
- FC SAN：在桌面虚拟化之后所需要的存储设备及交换机。
- Connection Broker：这也是一台安装 Connection Broker 的计算机硬件。



VMware View Server and Storage Infrastructure Configuration	Units	Unit Cost	Total Cost for the VMware View Solution
ESX server hardware	11	Y99,821.55	Y1,098,037
VC servers	1	Y8,922.44	Y8,922
FC SAN switch ports	11	Y20,590.25	Y226,493
FC SAN storage capacity cost	2,686	Y102.95	Y276,561
Connection broker server hardware	1	Y8,922.44	Y8,922
Total server and storage infrastructure capital expenditure for VMware View (implementing for all desktops - as per end of year 3 plans)			**Y1,618,936**

VMware View - Server and Storage Infrastructure Costs	Initial	Year 2	Year 3
Total cost	Y1,618,936	Y80,947	Y84,997
Schedule for virtualization	100.0%	0.0%	0.0%
Scheduled total cost	**Y1,618,936**	**Y0**	**Y0**

▲ 桌面虚拟化在服务器的成本上相当高

结 语

企业进入虚拟化之前的所有问题，在本章中都获得了基本的解答。然而虚拟化是一条漫长而复杂的路，不同的企业规模和期望的结果，都会直接影响虚拟化中的花费和整体回报。在真正开始虚拟化之前，妥善的规划，利用 VMware 或 Hyper-V 所提供的 TCO/ROI 计算器来做成本的计算及评估，相信在企业的虚拟化之路能避开更多的错误，并且充分利用虚拟化所带来的便利。

读书笔记

第 8 章 安装 vSphere 的基础：ESX/ESXi

关键词：

- 配置服务器的 BIOS 支持 ESX
- 配置 RAID
- 配置 KVM 方便操作
- 在物理机安装 ESX/ESXi
- 在 VMware Workstation 上安装 ESX/ESXi

安装 ESX/ESXi 是 vSphere 的基础，我们在前面的几个章节已经打好完整的基础，接下来就是进行安装了。本章将说明在物理机安装 ESX/ESXi 的准备任务，我们也会说明如何在虚拟机中安装 ESX/ESXi。

8.1 准备安装 ESX/ESXi 的服务器

安装 ESX/ESXi 就像安装任何一个 Linux 的套件一样，在确定你的机器匹配 VMware 的 HCL 之后即可以放心开始安装。但在安装之前，还是有些准备任务，包括了 BIOS 的配置，磁盘安全机制 RAID，以及远程键盘鼠标屏幕 KVM 的配置。



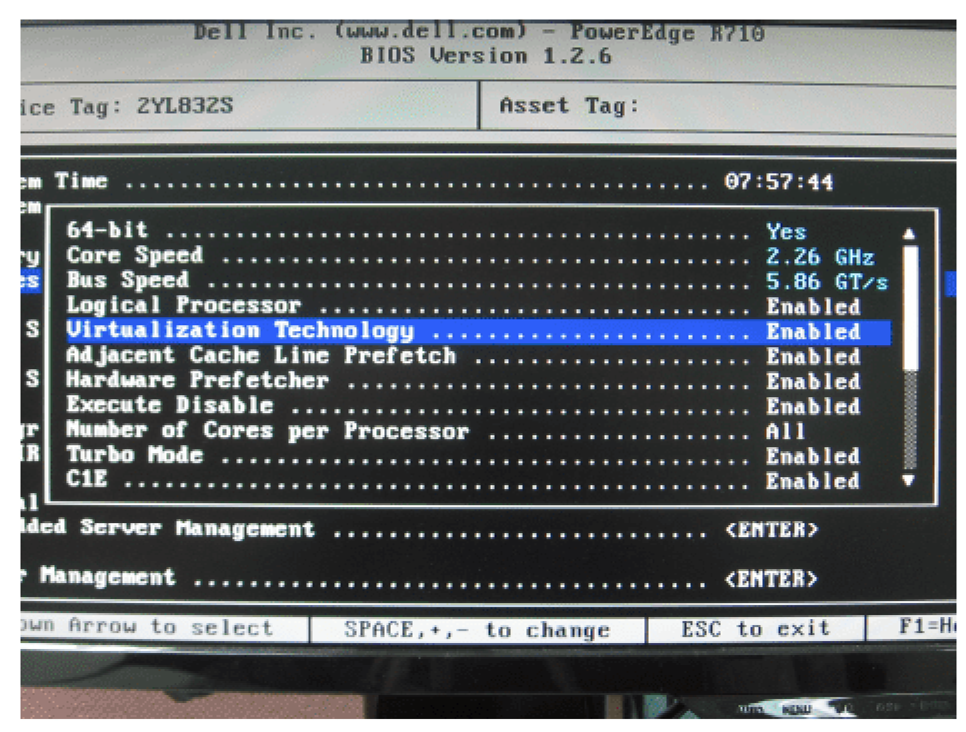
▲ 找一台匹配 HCL 且功能强大的服务器，会让 ESX 的安装更顺利

8.1.1 准备服务器的 BIOS 配置

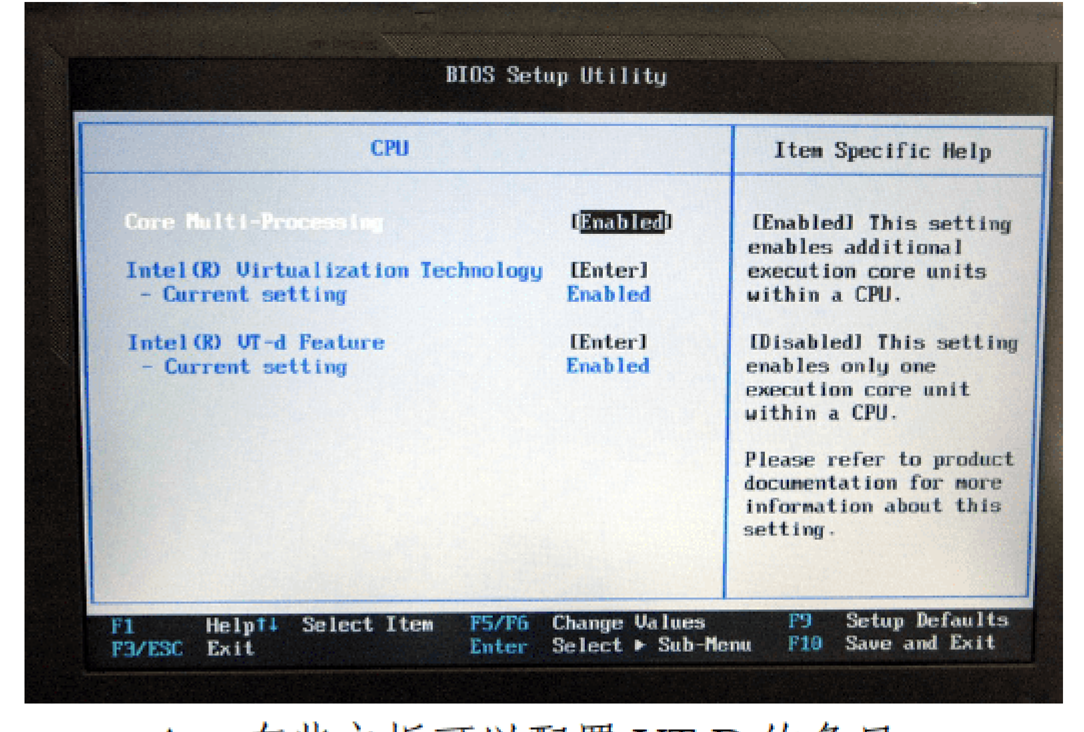
一般来说，任何通过 VMware HCL 认识的服务器都可以顺利运行，但在安装 ESX/ESXi 之前，还是别忘了将 BIOS 中相关的地方激活。

▶ 服务器 BIOS 中虚拟机选项的激活

- VT-X：在 BIOS 会显示 Virtualization Technology。
- VT-D：在 BIOS 中会显示 VT-D 或是 IOMMU。



▲ 这是 Dell R710 服务器配置 VT-X 的地方



▲ 有些主板可以配置 VT-D 的条目

8.1.2 RAID 的规划及安装

ESX/ESXi 虽然是一个 Hypervisor，但也是一个操作系统，因此也需要有些安全防护，因此在安装前，先要规划 Hypervisor 的存放位置，本地硬盘则一定要使用 RAID 才能确保 Hypervisor 的常规运行。

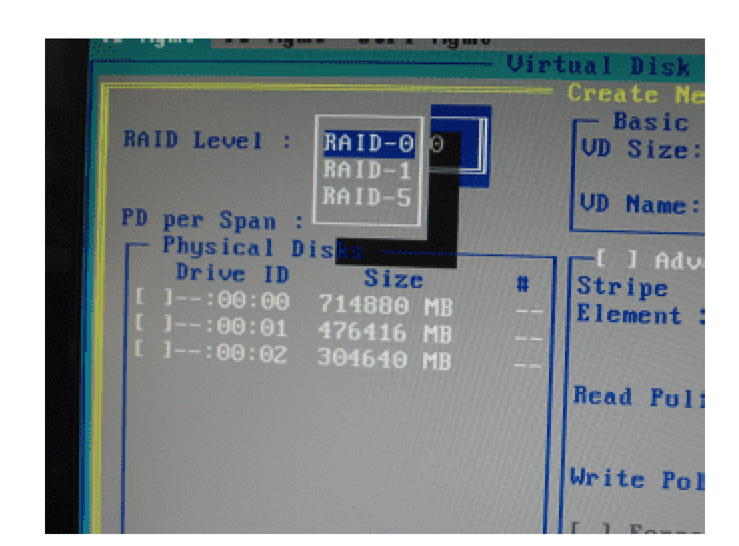
1. 选择哪一种 RAID

本地硬盘安装 Hypervisor 最常用的就是 RAID1、RAID10 或 RAID5。如果只有 2 个硬盘，使用 RAID 1 镜像是最方便的，如果有 3 个以上硬盘，可以做成 RAID 5 或是 RAID 1+Hot Spare，4 个则可以做成 RAID5 或 RAID10。有关 RAID 的详细说明，请参阅本书有关存储的章节。

在不同的服务器品牌中，对 RAID 有不同的名称。下面是几个最常见的名称：

▶ RAID 常见名称

- Volume：卷，即 RAID 落实后的组合。
- Virtual Device：和卷同意，但一个 VD 可以只有一台硬盘。
- IM：即 RAID1，通常是 Mirror 的意思。
- IS：即 RAID0，即 Stripping 的意思。
- IME：即 IM+Enhance 的意思，通常为 RAID1+How Spare。
- PD：Physical Device，即物理硬盘。



▲ 不同的 RAID 有不同的目的，但本地最常用的就是 RAID1/5

2. 规划 RAID 1 演示：使用 IBM 内置 LSI

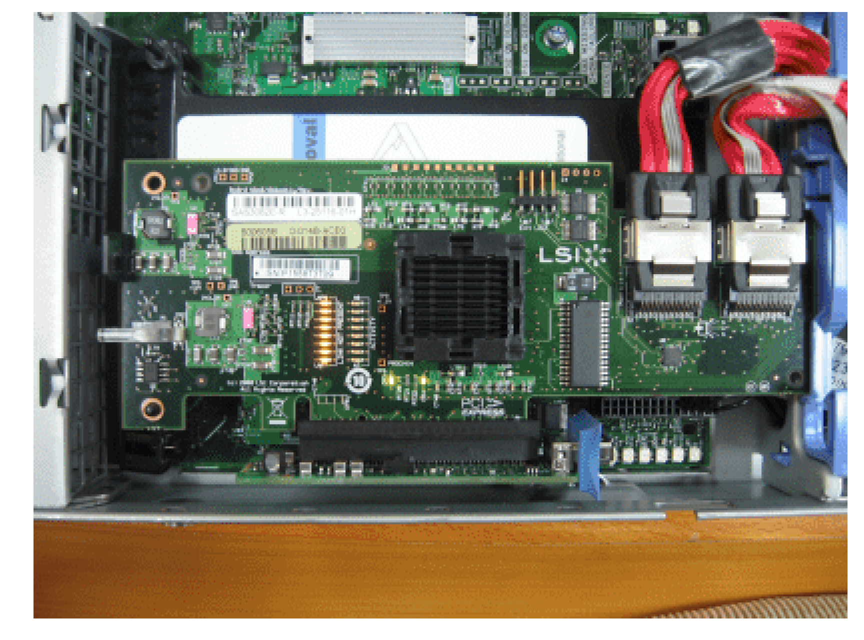
这次演示使用的服务器是 IBM 的 X3550 M2，使用 SAS 界面的 2.5 英寸硬盘，本身也有 SR-BR10i RAID 总控卡，可以规划成不同的 RAID，虽然没有 Dell Perc 6/i 的功能多，但安装 ESX/ESXi 也算足够，我们就以这个总控卡来演示创建 RAID 1 的磁盘组合。



▲ IBM 的服务器当前多使用 2.5 英寸的硬盘



▲ 也支持热插拔十分方便



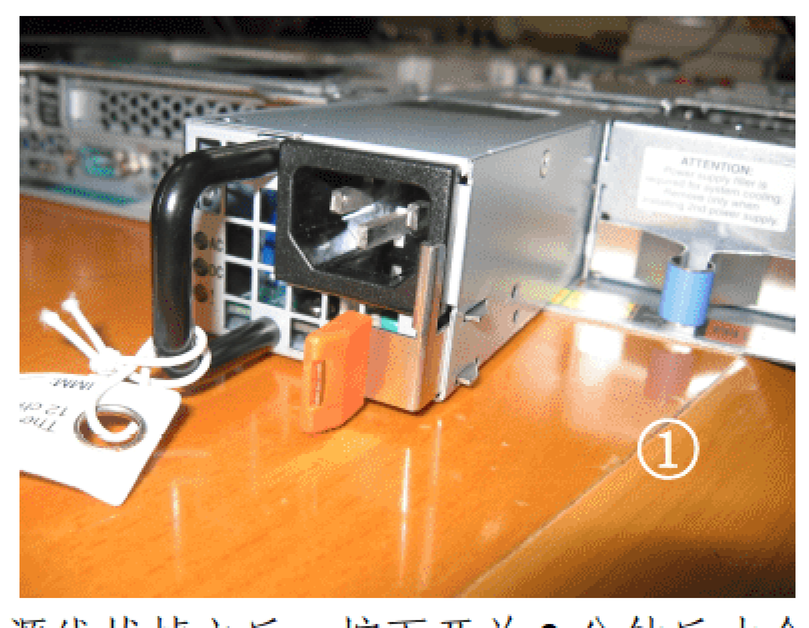
▲ 这是 IBM X3550 M2 使用的 LSI RAID 卡，支持 SATA 和 SAS

3. 使用 IBM SR-BR10i RAID 卡创建 RAID 1

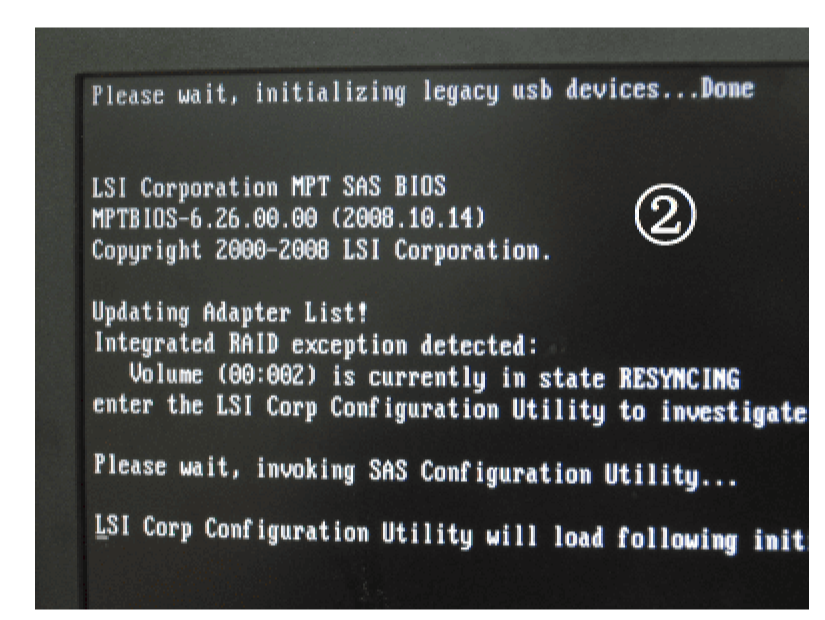
1. 引导进入 IBM X3550 M2 的机器中，要注意，如果电源线被拔掉之后引导，要等 3

分钟服务器才会激活。

2. 当屏幕弹出 BIOS 画面，等到弹出 RAID 的配置画面时，按 Ctrl+C 组合键进入 RAID 的配置画面。



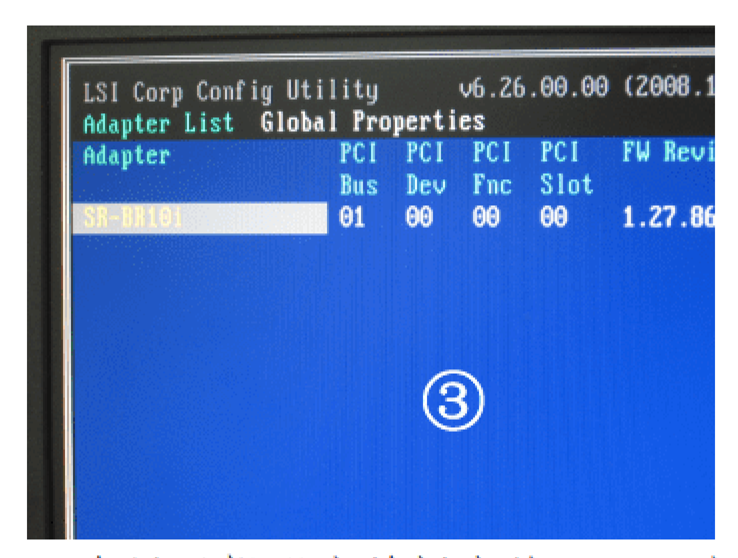
▲ 若电源线拔掉之后，按下开关 3 分钟后才会激活



▲ 进入 RAID 的配置画面

3. 此时会弹出机器所使用的 RAID 卡，就是 SR-BR10i。按 Enter 键进入选项。

4. 我们先选择硬盘的配置，选择 SAS Topology 选项。



▲ 会显示机器当前插上的 RAID 卡



▲ 先进入 SAS 硬盘的配置来看有几个硬盘

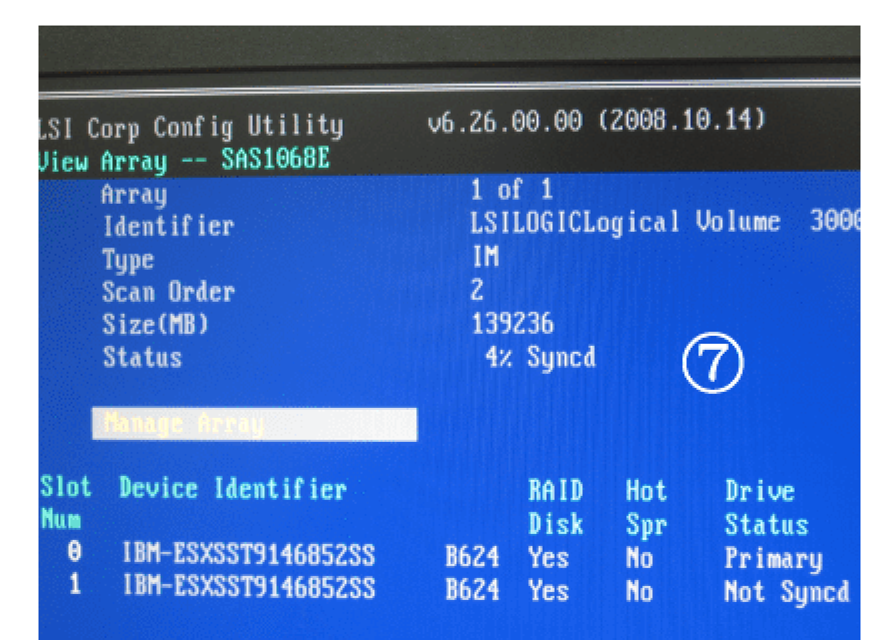
5. 这时可以看到服务器中所有的物理硬盘，左下图中可以看到在 Phy 物理硬盘都已经被配置成 RAID 的群组了，这里将物理硬盘配置成 RAID 群组。

6. 回到上一层菜单，并且选择 RAID Properties 选项。

7. 在这个画面，可以看到要创建的 RAID。我们直接按 C 键就可以创建一个新的 RAID 群组。



▲ 图中有物理硬盘和已经做成 RAID 的硬盘



▲ 在这个画面按 C 键就可直接创建新的 RAID 群组

8. 按 C 键之后，系统会询问 RAID 的种类，在其中要选择 IM，意味着是 RAID 1。选择完毕之后即按下存盘即可。

9. 可以看到已经创建好一个 RAID 1 群组了。



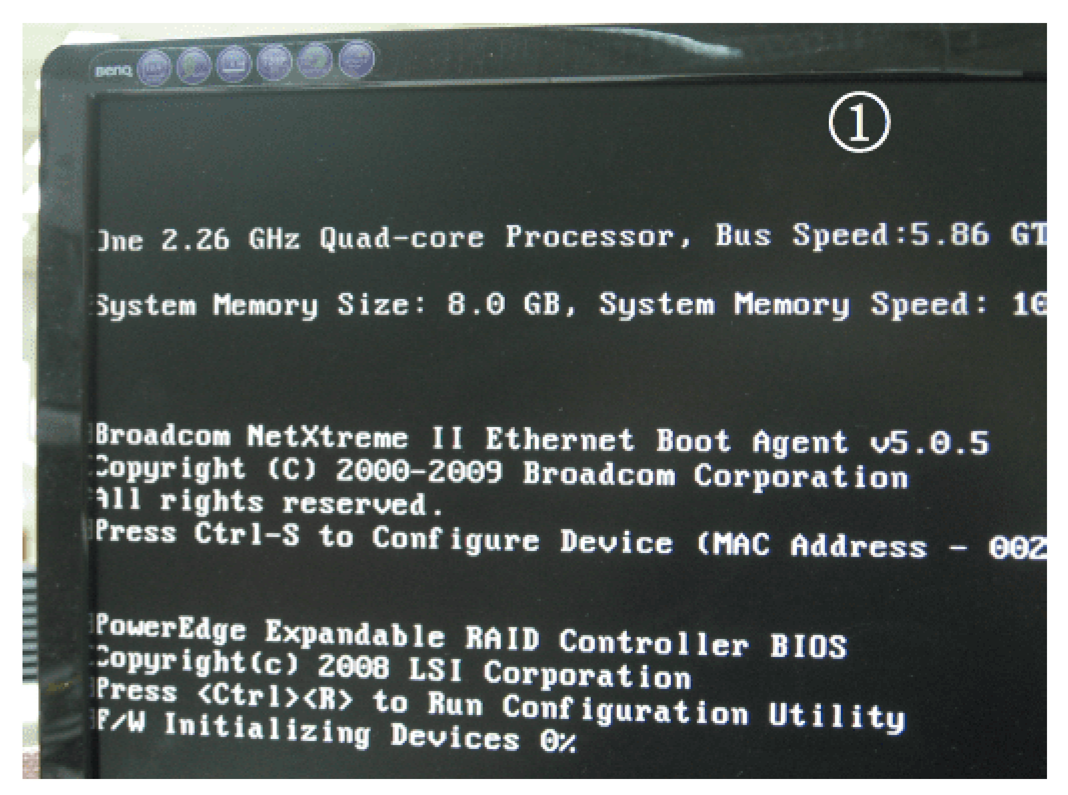
▲ 可以看到已经将两个硬盘创建成 RAID 1 了

4. 规划 RAID 5 演示：使用 Dell Perc 6/i

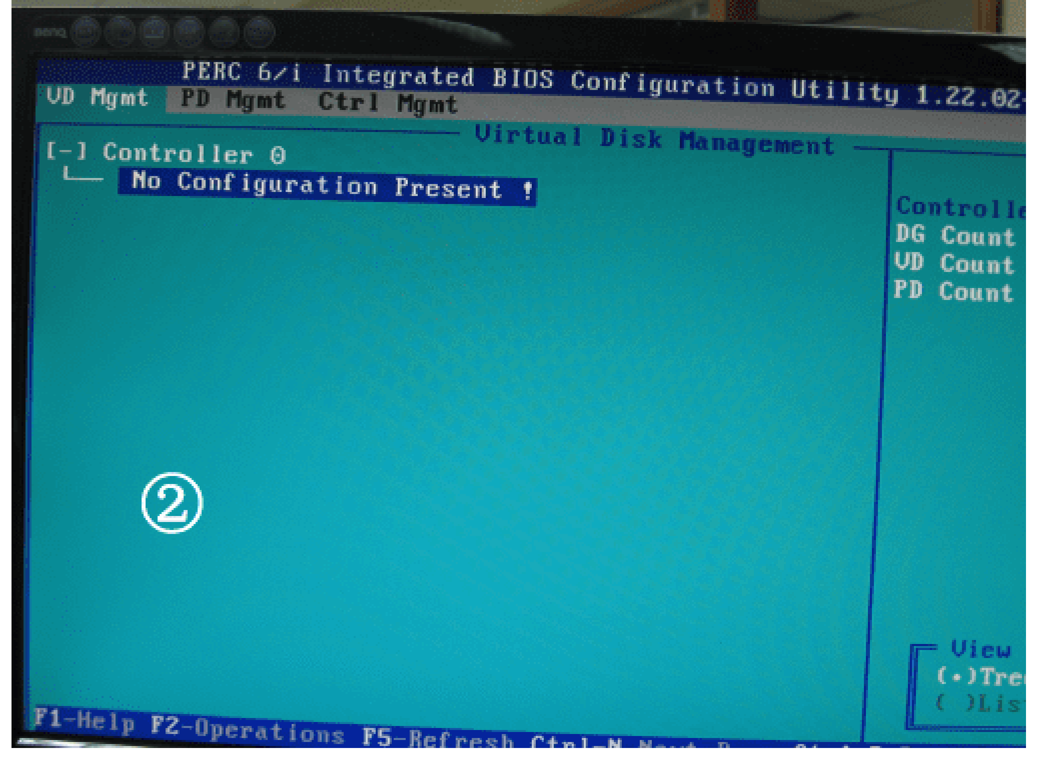
Perc 6/i 是 Dell 服务器最常见的存储界面，可支持 SAS 或是 SATA 硬盘，并且支持各种的 RAID。我们就以 Perc 6/i 来演示规划 RAID 5。

▶ 使用 Dell Perc 6/i 规划 RAID 5

1. 先将 Dell 主机引导，当弹出右下图中 RAID 配置画面时，按 Ctrl+R 组合键。
2. 此时会进入 RAID 的配置画面，左下图中会先显示当前使用的 RAID 卡。

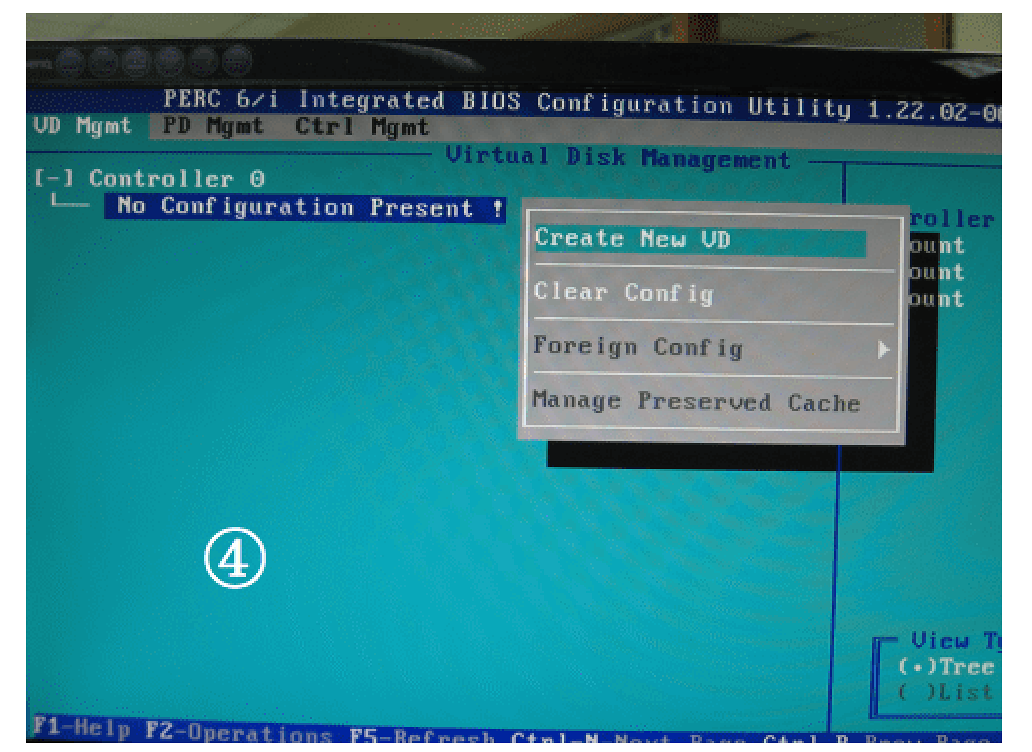


▲ 按 Ctrl+R 组合键

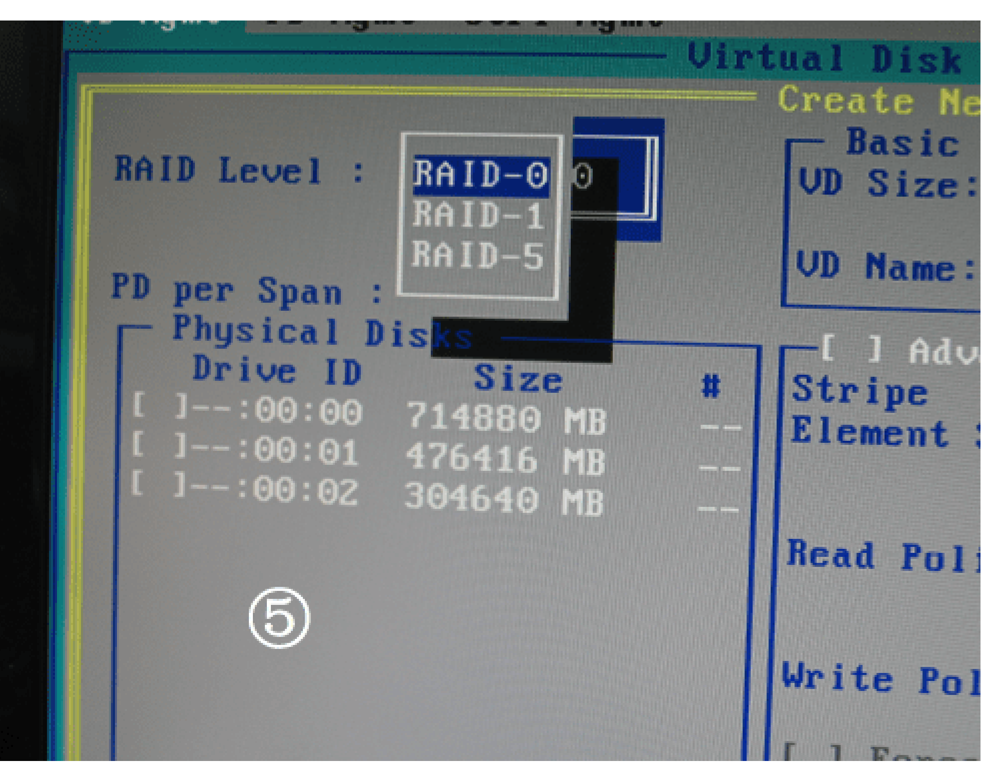


▲ 给出当前使用的 RAID 卡，还没有任何配置

3. 如果这台服务器还没有做过任何 RAID，可以直接按 Enter 键进入 RAID 配置画面，或是按 F2 键选择创建。
4. 在 Perc 6/i 中，不管是不是 RAID，都得创建一个 Virtual Disk，简称 VD。这里的 VD 和 ESX/ESXi 中的虚拟硬盘是不一样的东西。VD 就是硬盘组合，从 1 台到 6 台都可以组成一个 VD。创建 VD 就是创建 RAID。
5. 在进入创建 VD 画面后，可以看到不同的选择，Perc 6/i 支持 RAID 0/1/5，我们选择 RAID 5。



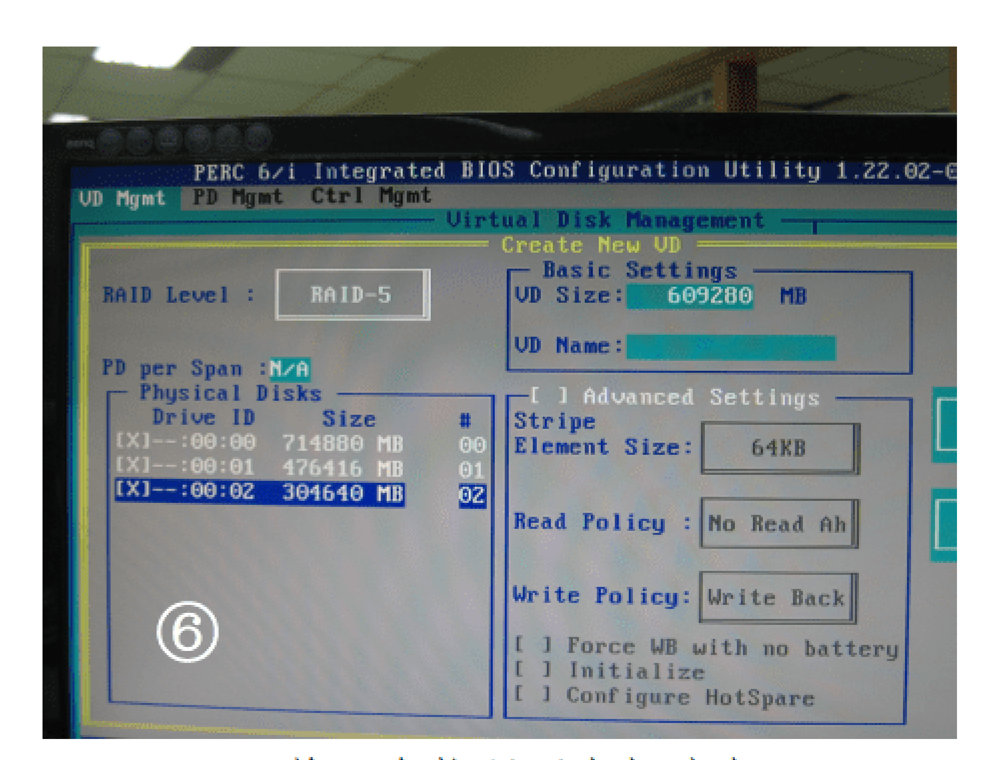
▲ 按 F2 键之后可以选择创建一个新的 RAID 组合，称为 VD



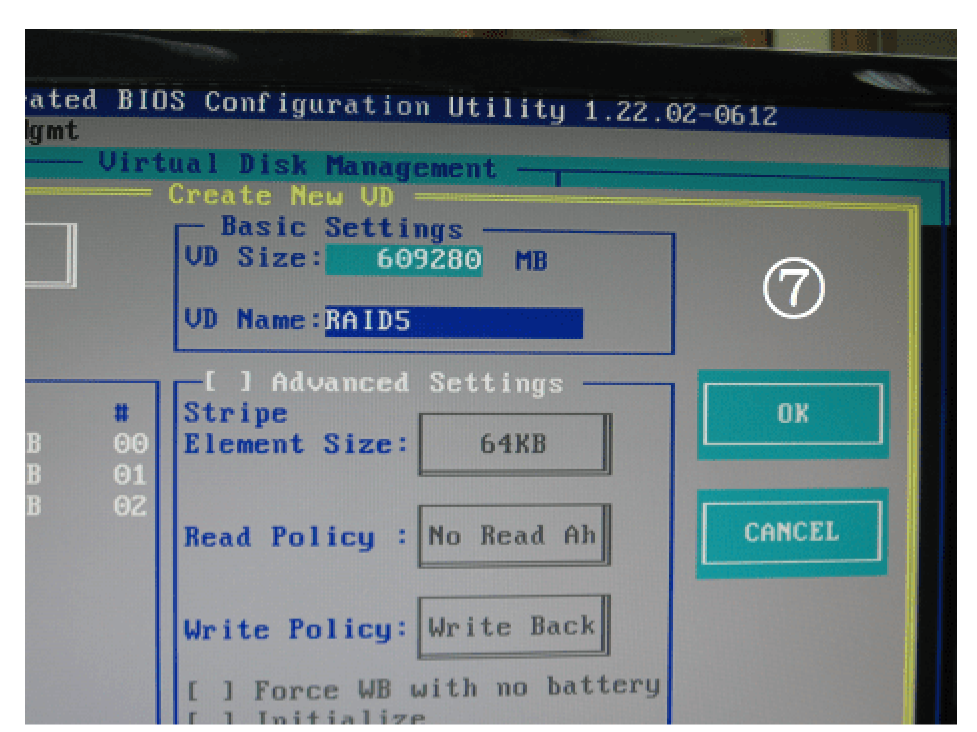
▲ 支持 RAID 0/1/5，我们选择 RAID 5

6. RAID5 需要至少 3 个硬盘，下方列表会给出当前可以加入 RAID 的物理硬盘，就把这 3 个硬盘都选中。使用 TAB 切换条目，使用 Space 来选择。

7. 选中之后，系统会自动算出这个 RAID 5 组合能有的空间大小，并且要求你键入一个 RAID 5 的名称，我们就键入 RAID 5。



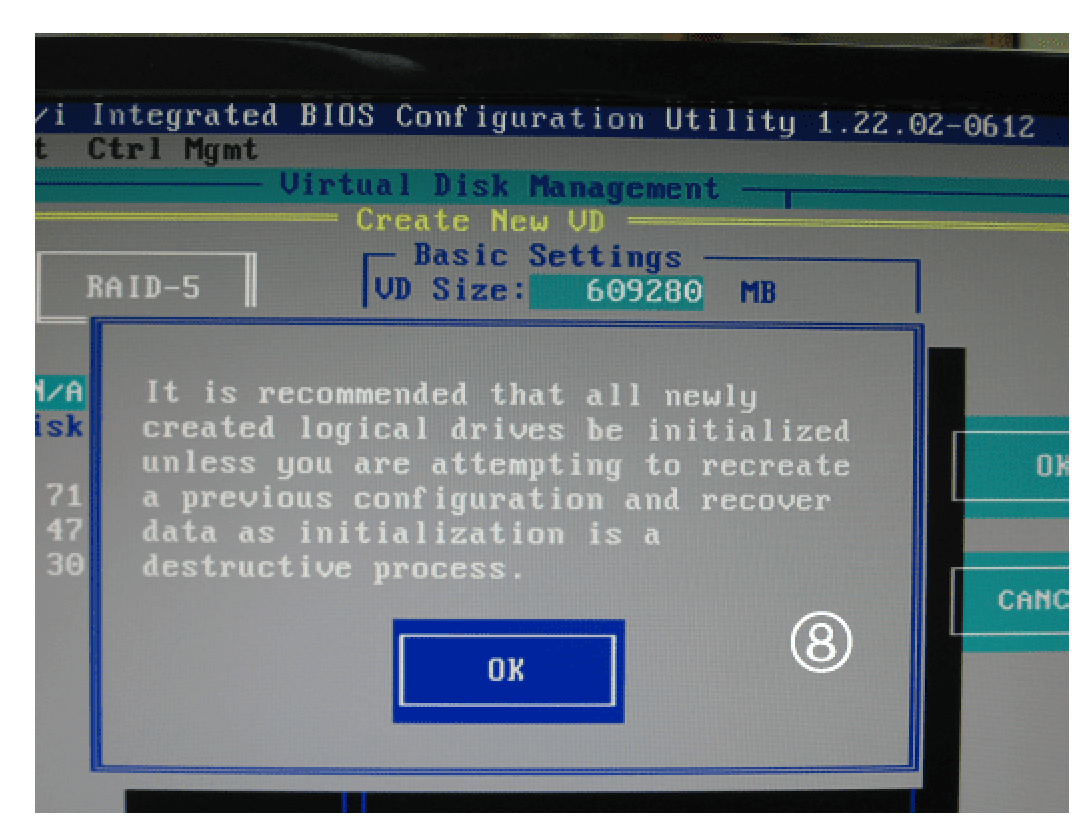
▲ 将 3 个物理硬盘都选上



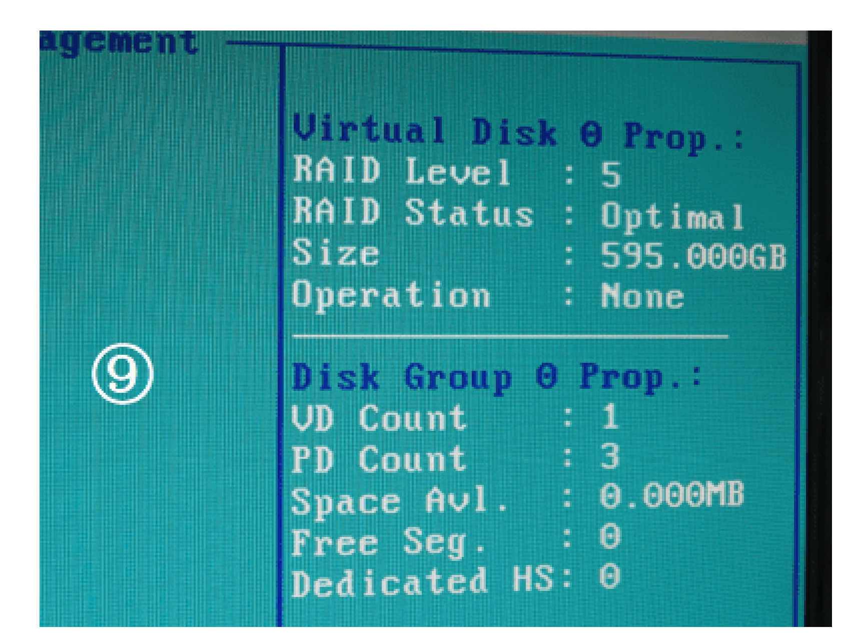
▲ 此时系统会自动算出大小，在此键入 VD 的名称

8. 之后单击 OK 按钮，系统会弹出确定画面，再单击 OK 按钮。

9. 接下来我们可以看到 RAID 5 已经创建完毕了。



▲ 系统会将硬盘中的数据都删除



▲ RAID 5 已经创建完毕了

10. 确定离开后，这 3 个物理硬盘创建的 RAID 5 就会落实，接下来重启。引导后可以在

BIOS 中看到已经创建了 RAID 了。

```
One 2.26 GHz Quad-core Processor, Bus Speed:5.86 GT/
System Memory Size: 8.0 GB, System Memory Speed: 106

Broadcom NetXtreme II Ethernet Boot Agent v5.0.5
Copyright (C) 2000-2009 Broadcom Corporation
All rights reserved.
Press Ctrl-S to Configure Device (MAC Address - 0024

PowerEdge Expandable RAID Controller BIOS
Copyright(c) 2008 LSI Corporation
Press <Ctrl><R> to Run Configuration Utility
HA -0 (Bus 3 Dev 0) PERC 6/i Integrated
FW package: 6.2.0-0013

1 Virtual Drive(s) found on the host adapter.

1 Virtual Drive(s) handled by BIOS
```

⑩

▲ 可以看到已经有创建一个 VD，就是 RAID 5 了

8.1.3 准备主机的总控 KVM

在机房环境中，如果每一次安装或配置服务器都要跑到计算机旁边，安装上键盘、鼠标、屏幕，不但麻烦，一不小心还会把服务器的电源或网线给碰掉，带来无谓的困扰。在 IDC 中，一般会使用所谓的 KVM 转移设备，将这台服务器的键盘、鼠标、屏幕（通称 KVM）的输出入都转移出来。



▲ 小型 SOHO 环境可以准备一个多切换器，但机房环境就无法使用了

1. KVM 的种类：交换机

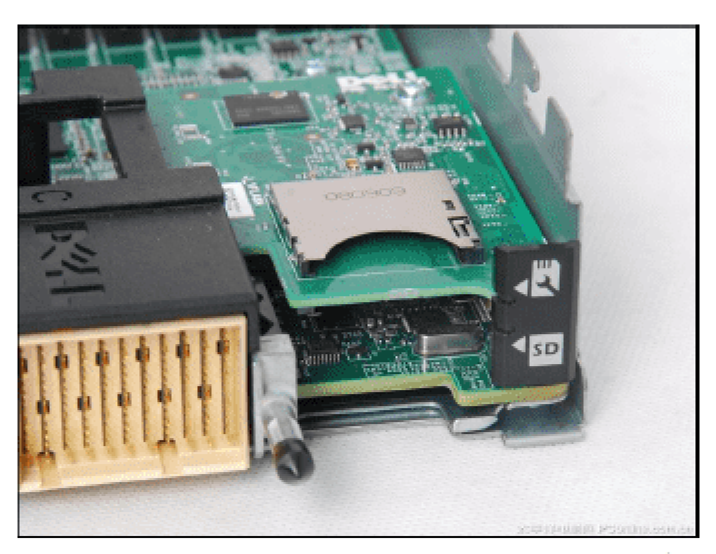
最常见的 KVM 转移设备就是网卡，一般做法有通过 KVM 的交换机，再利用客户端程序或浏览器连上交换机来操作给定 IP 的服务器。KVM 交换机通常会有一个屏幕界面和 USB 界面，并且将这些界面集成 CAT 网络界面。



▲ KVM 交换机是将所有 KVM 导入一个网络口，再接上公用交换机

2. KVM 的种类：专属总控卡

另一种就是在服务器中直接插上网络界面的 KVM over IP 卡，可以将这个 KVM over IP 的卡配置一个私有的 IP，再将这个卡连上公用的交换机，只要通过 IP 就可以直接操作这台计算机，大部分的服务器产品都是使用这种方式。



▲ 一般是以界面卡的方式插在主机上，并且通过网络 IP 来总控

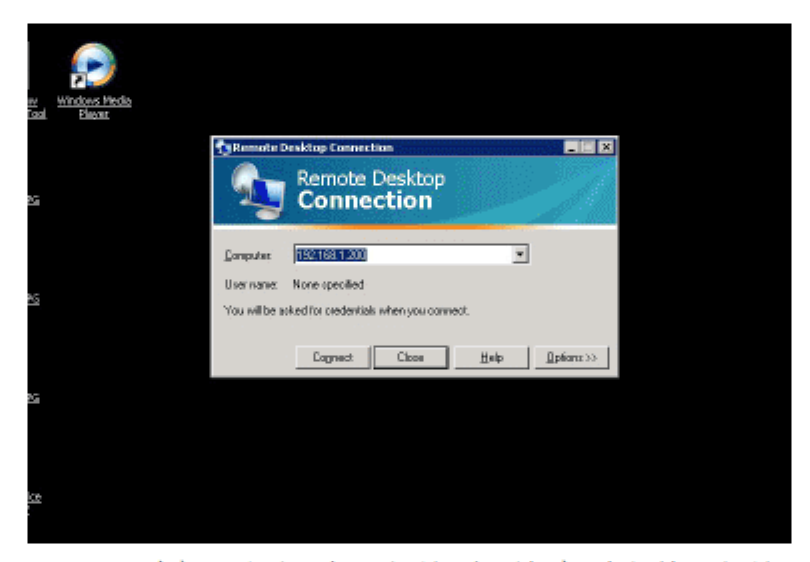


▲ 在 IBM 的主机上称为 System Management Port

注意

KVM over IP 和远程联机

很多读者一定会问，使用远程桌面不也可以总控计算机？但远程桌面是一个需要已经安装好操作系统的计算机才能使用的服务，但 KVM 则是计算机一引导就立马可以接手总控，并且可以配置 BIOS、RAID 等，等于就是在计算机旁边一样，是让 IT 人员事半功倍的好帮手。



▲ 远程联机必须主系统安装好操作系统才行使用

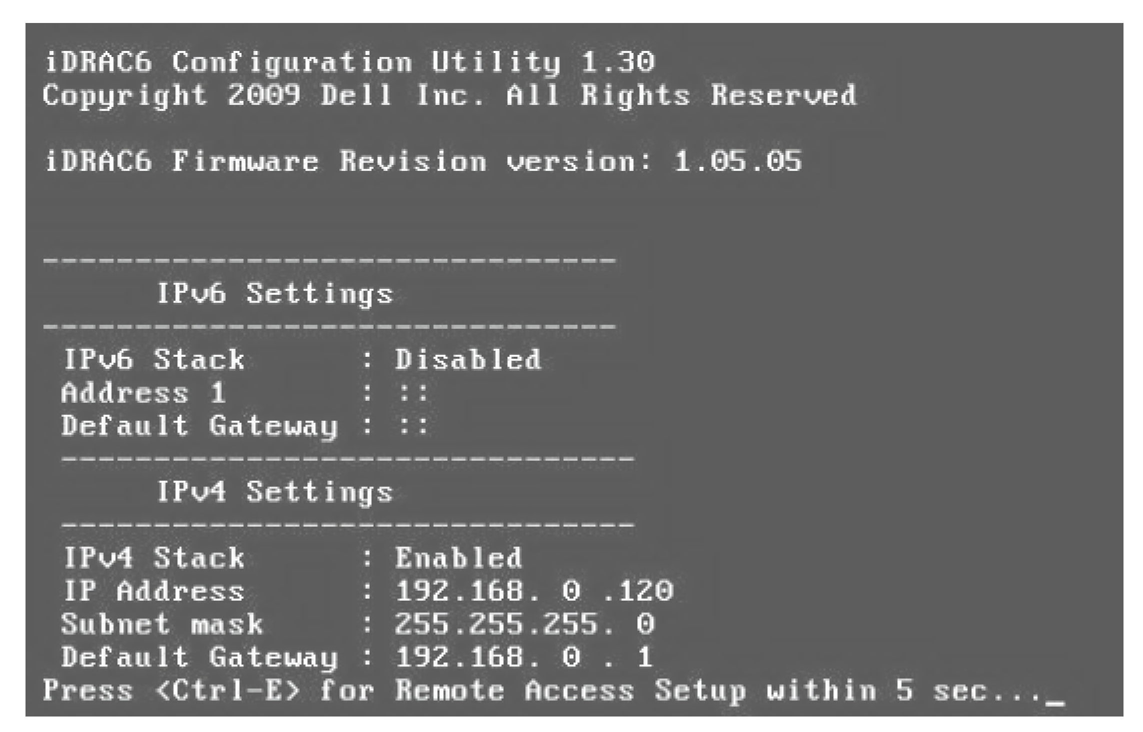


▲ KVM 是主机一引导就可以总控了

▶ 使用 KVM：以 Dell iDrac6 Enterprise 为例

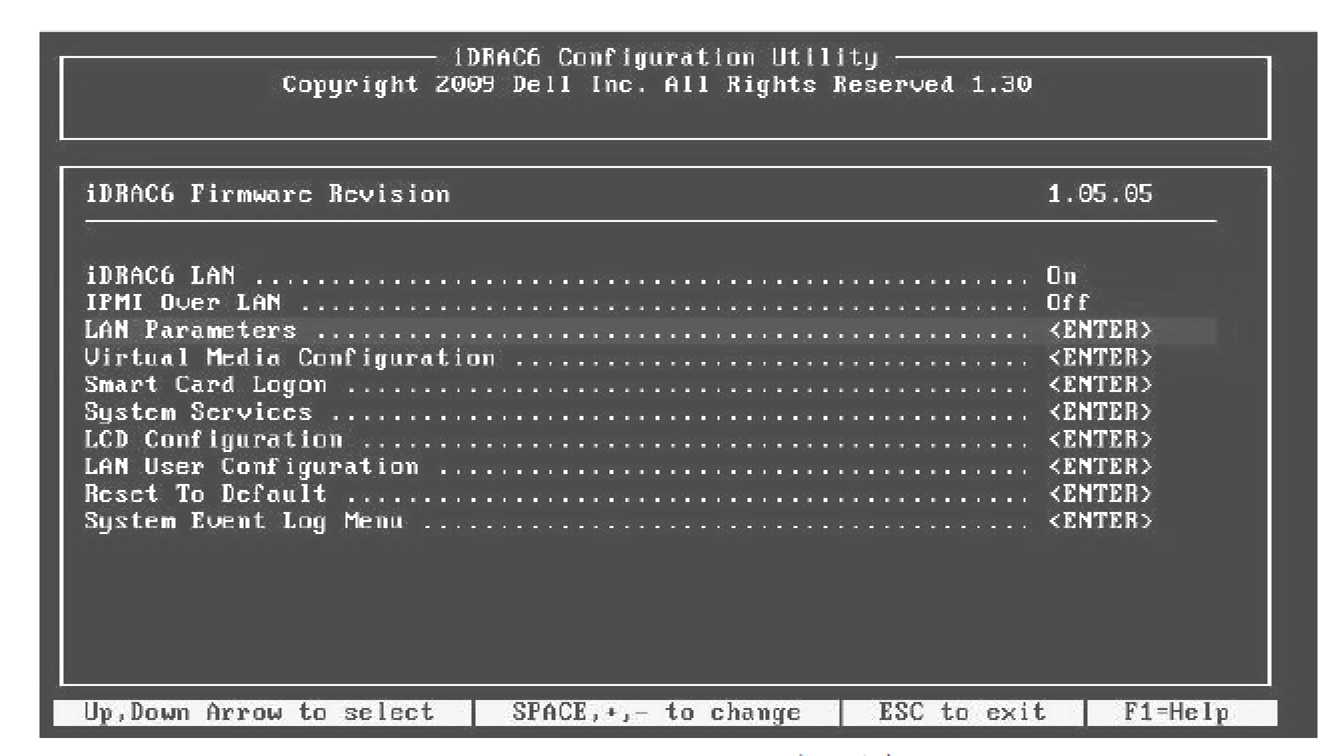
我们就以 Dell 服务器上的 iDrac6 Enterprise 为例，这是一片 PCIE 界面的 KVM overIP 的界面卡，在插上之后，服务器背板上就会有一个网络口，从这个网络接上公用交换机，就可以通过 IP 来总控 KVM 了。接下来就来配置 KVM over IP。

1. 配置 iDrac6 Enterprise 最重要的就是配置其 IP。在安装好之后，默认的 IP 网段是 192.168.0.X。我们在 Dell 的计算机引导后，按 Ctrl+E 组合键进入 BIOS 配置选项。



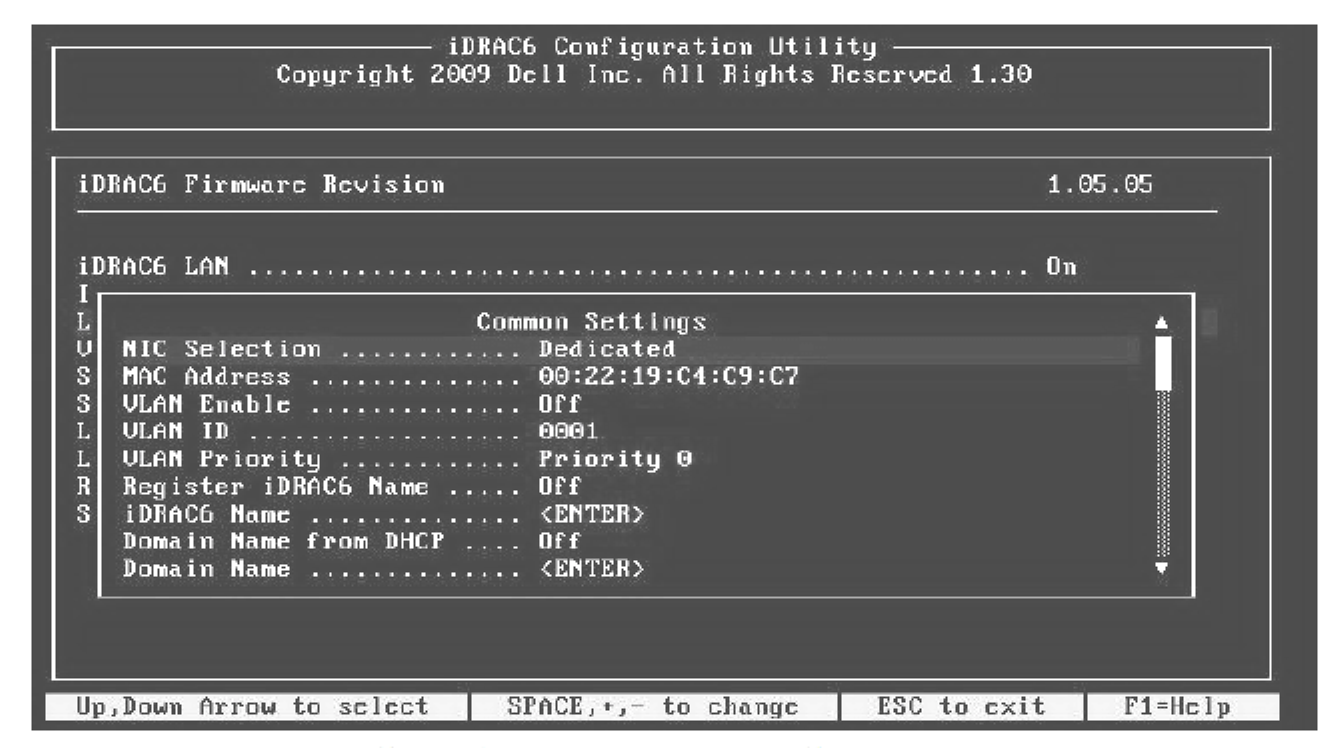
▲ 配置 KVM 的 IP 是最重要的，按 Ctrl+E 组合键可以从 BIOS 进入配置

2. 进入之后，要注意 iDrac6 LAN 一定要 On，接下选择 LAN Parameters 选项。



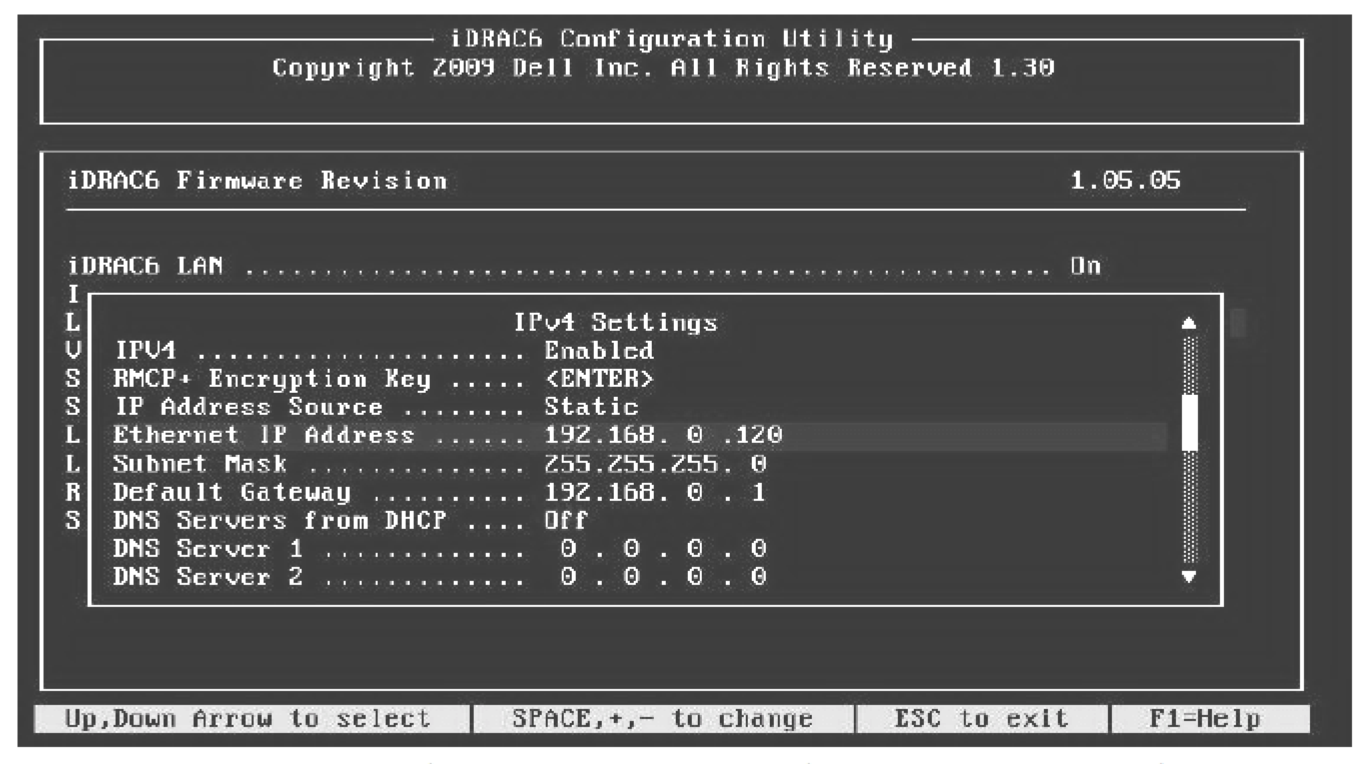
▲ iDrac6 LAN 一定要打开

3. 接下来先将 NIC Selection 配置为 Dedicate。iDrac6 Enterprise 可以使用主板上的网卡（LAN on Motherboard，LOM）作为通信的网络口，但只要 ESX/ESXi 一装载网络的驱动程序就无法使用了。



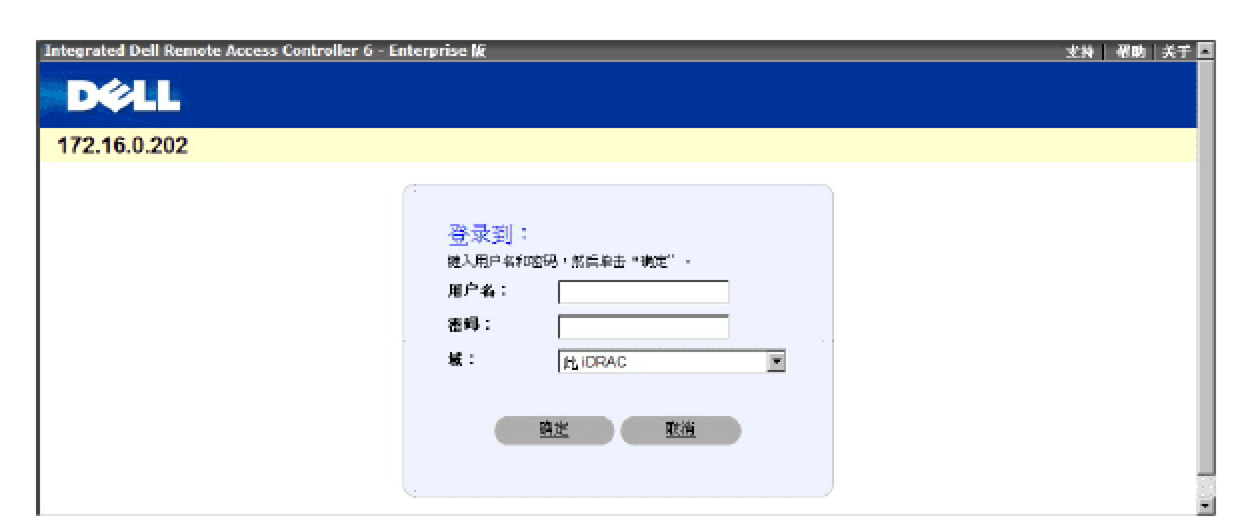
▲ 可以使用专用卡，也可以使用主板上的网卡

4. 接下来是配置 iDrac6 Enterprise 的 IP，默认值是 192.168.0.120，如果要接到公共的交换机上，则要改成和公共交换机一样的网段，也可以使用 DHCP。



▲ 配置 IP，最好配置和公网交换机上一样的 IP 才好操作

5. 当配置好 IP 之后，从浏览器键入 IP，就可以连入 iDrac6 Enterprise 的界面了。其中有很多选项，但我们最在乎的还是远程连入。



▲ 登录 iDrac6 的界面，为简体汉语



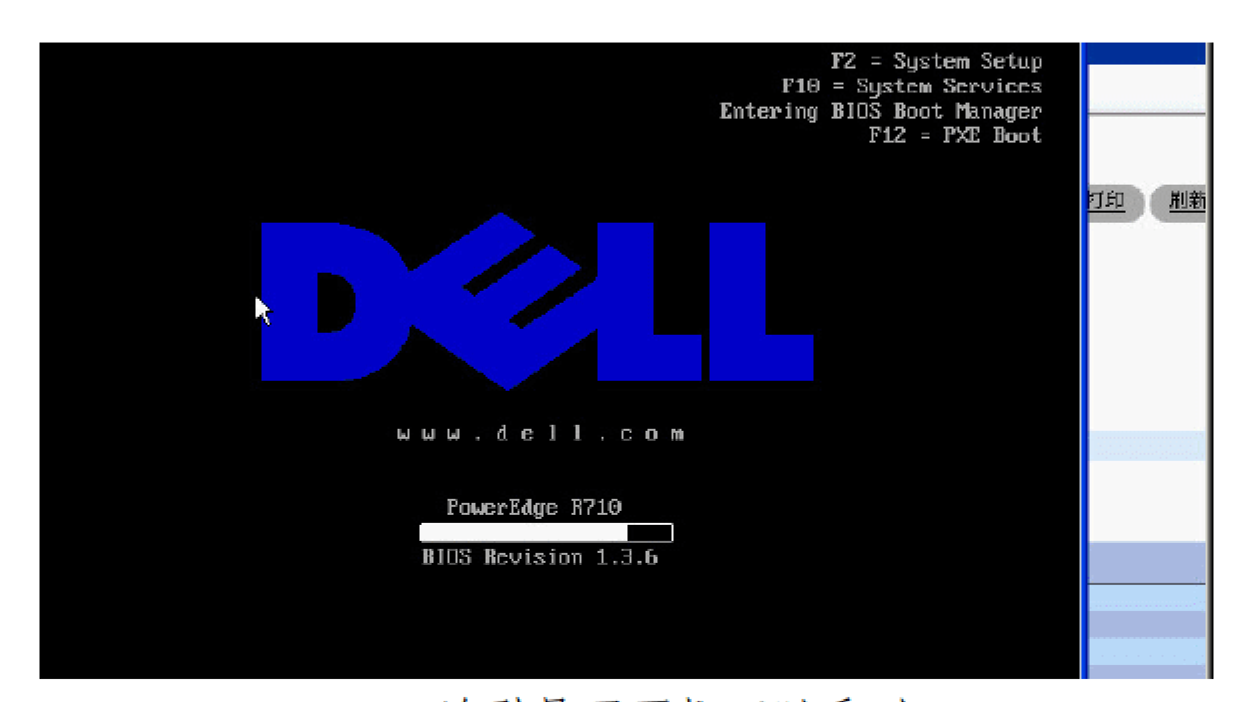
▲ 功能很强大，要安装时还是选择“远程访问”

6. 选择“远程访问”（iDrac6 为简体汉语界面），然后选择“激活 Viewer”，系统会开一个新窗口，可以直接总控服务器。



▲ 激活 Viewer 之后就可以打开远程 KVM 总控了

7. 直接总控服务器的画面，连引导画面及 BIOS 配置都可以完整访问。



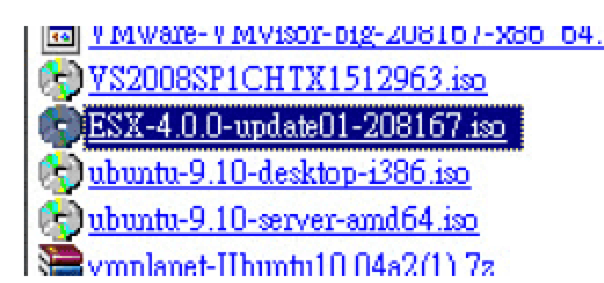
▲ 连引导画面都可以看到

8.2 开始安装 VMware ESX/ESXi

在 BIOS 配置好，RAID 创建完毕，KVM 也可以总控之后，我们就可以从远程直接安装 ESX 了。在这一小节，我们除了介绍在服务器上安装 ESX 之外，也会介绍在 VMware Workstation 下安装 ESX 作为测试用途。

8.2.1 在物理机上安装 ESX4.0

在本书写作时最新的 VMware ESX 版本为 ESX4.0 Update 1，官方版本编号为“VMware ESX 4.0.0 build-208167”，安装前要先确定拿到最新的版本。接下来就是完整的安装步骤。



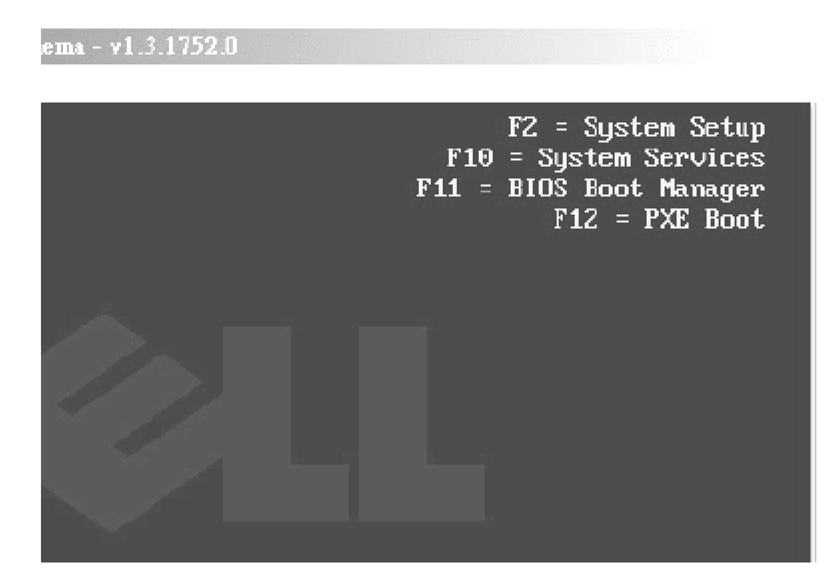
▲ 最新的版本是 4.0 Update1

1. 安装介质的选择

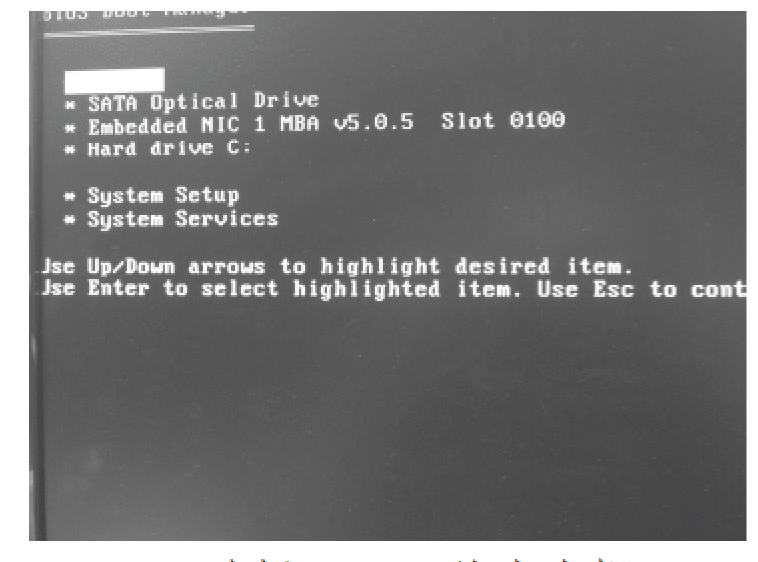
安装时，可以直接将 ESX 的光盘片到服务器的光盘机中，如果有使用 iDrac6 Enterprise 的话，可以将 ISO 文件从任何一台使用 iDrac6 的个人计算机上对应过去。

2. 安装 ESX 的光盘

1. 如果是使用光盘，直接放入服务器的光盘机中，并且激活服务器。本例使用 Dell R710 演示。
2. 引导后，按 F11 键可以选择引导时的介质。
3. 当弹出选择画面时，选择 SATA Optical Drive。就会进入光盘机。

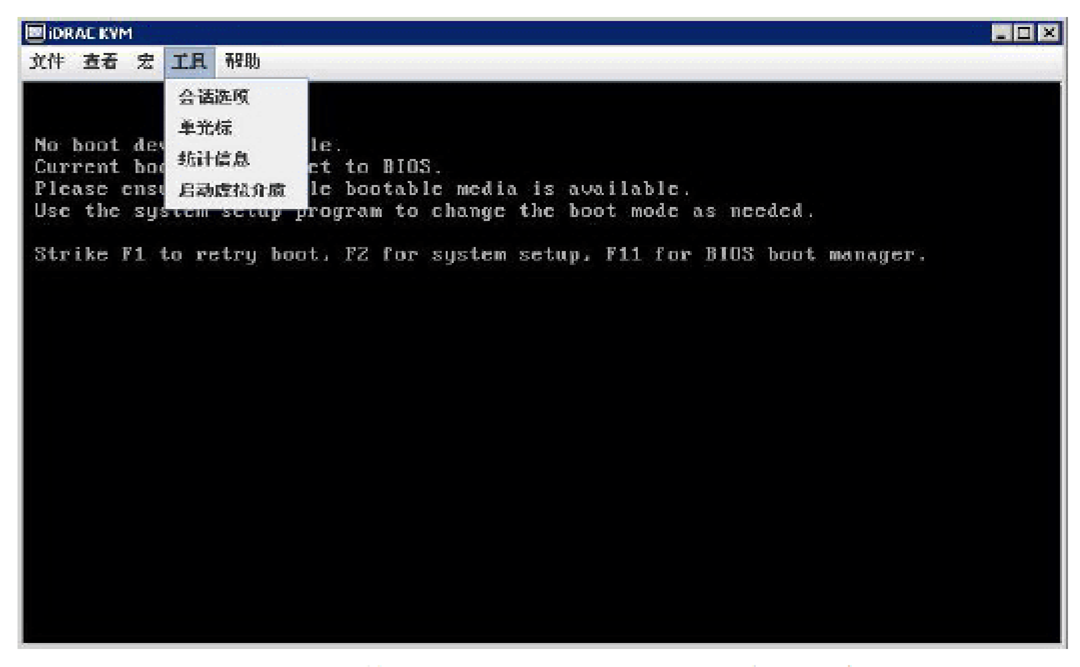


▲ 引导时按 F11 键可以选择引导介质



▲ 选择 SATA 的光盘机

4. 如果使用 iDrac6，激活 Viewer 之后，在引导之前，选择“工具”/“激活虚拟介质”选项。



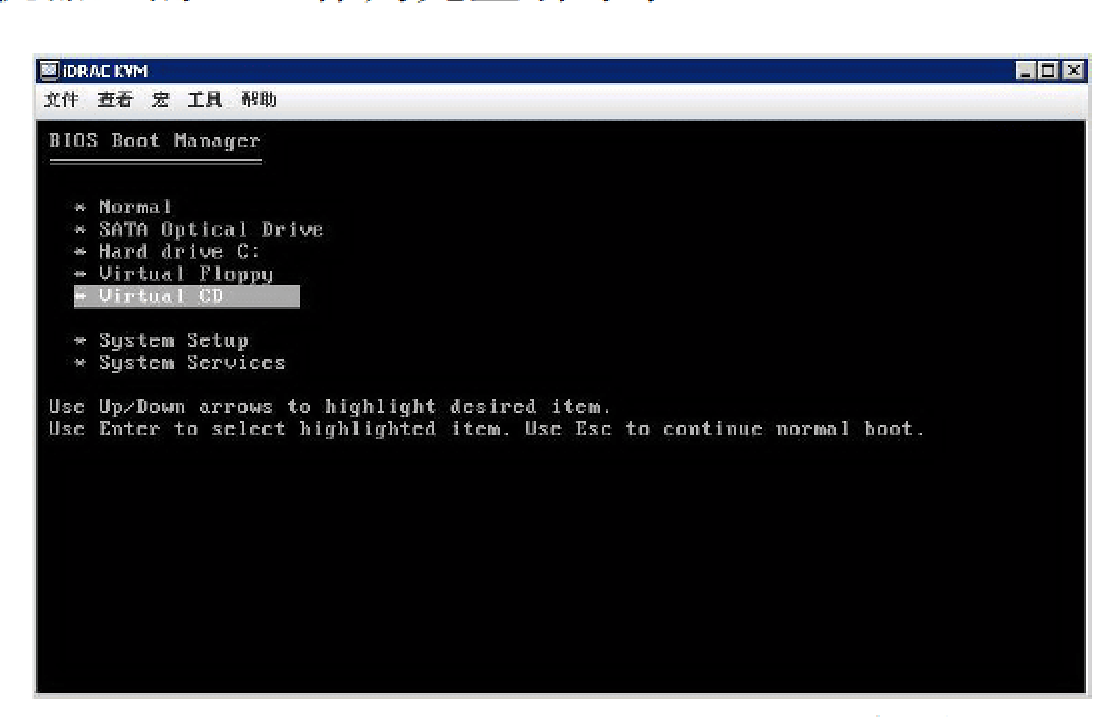
▲ 如果使用 iDrac6，选择虚拟介质

5. 此时会要求你对应一个本地的 ISO 文件，选择你的 ESX ISO 文件（要自行先制作成 ISO 文件），然后在“映射”的地方打勾。注意这个窗口不要关闭，要不然引导后会抓不到这个介质。



▲ 将 ISO 挂上之后，这个窗口不要关闭

6. 此时在引导画面中，按 F11 键选择引导选项，注意这时候就可以选择 Virtual CD，就会从你操作机器上的 ISO 作为光盘引导了。



▲ 选择 Virtual CD，就会从这个 ISO 档引导了

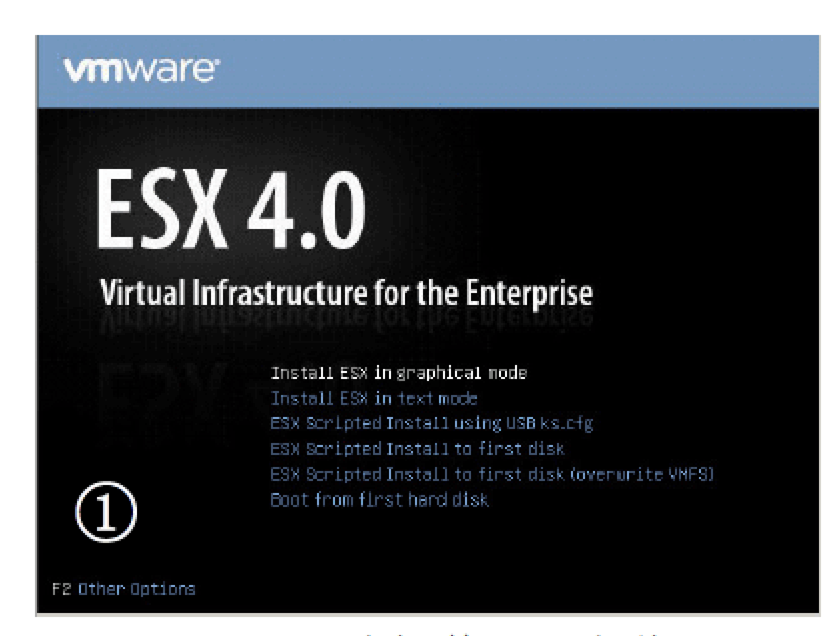
3. 开始安装 ESX

安装 ESX 和安装任何一个 Linux 没什么差别，整个过程十分简单。

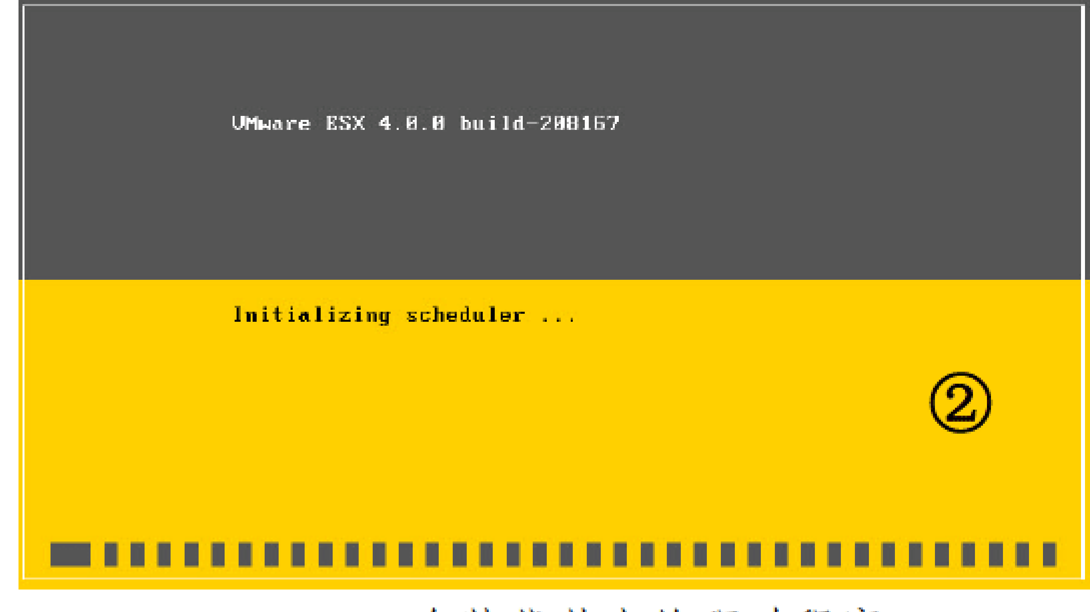
▶ 在物理服务器上安装 ESX

1．引导后，选择 Install ESX in graphical mode 选项。

2．此时会弹出标准 ESX 的装载画面，装载基本的驱动程序。



▲ 选择第一项安装



▲ 会装载基本的驱动程序

3．接着会装载安装的 Linux Console。

4．接着弹出欢迎画面，单击 Next 按钮。




▲ 标准的 Linux Console 装载



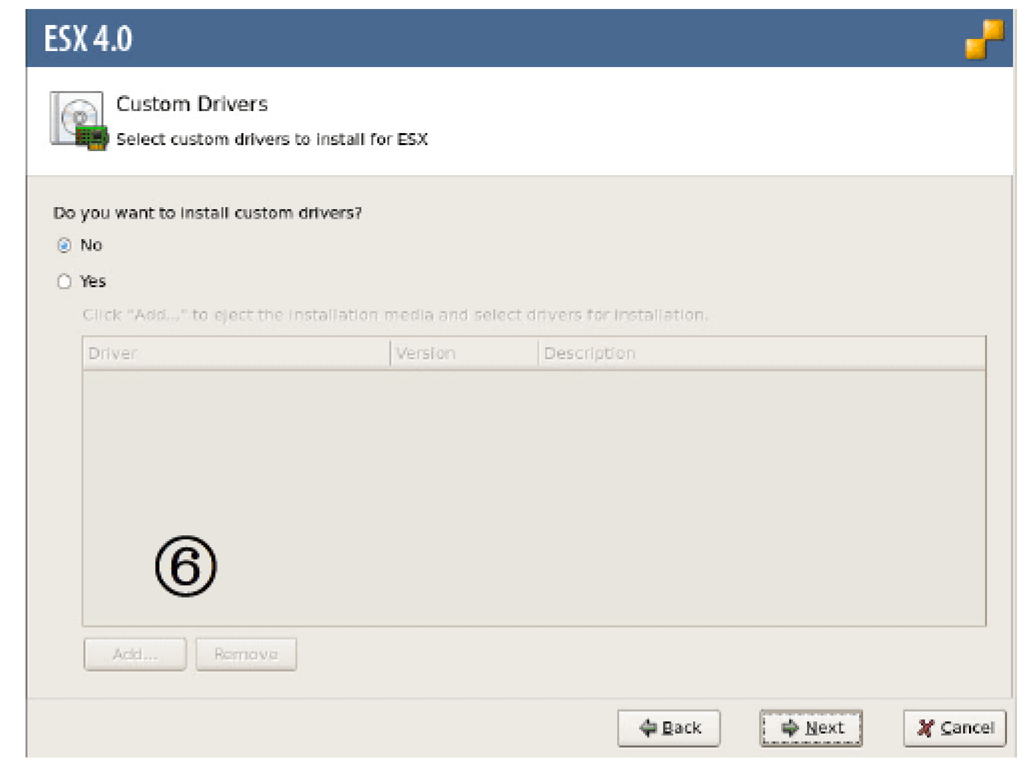
▲ 单击 Next 按钮

5．版权页，接受之后单击 Next 按钮。

6．在选择完 US 为键盘语系后，系统会询问是否需要安装额外的驱动程序，选择 No 之后单击 Next 按钮。



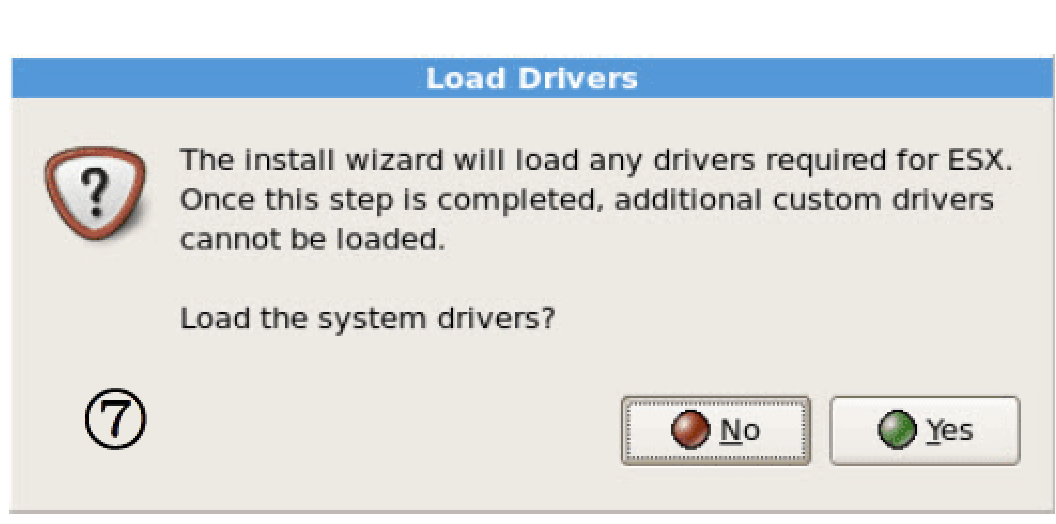
▲ 接受版权页后单击 Next 按钮



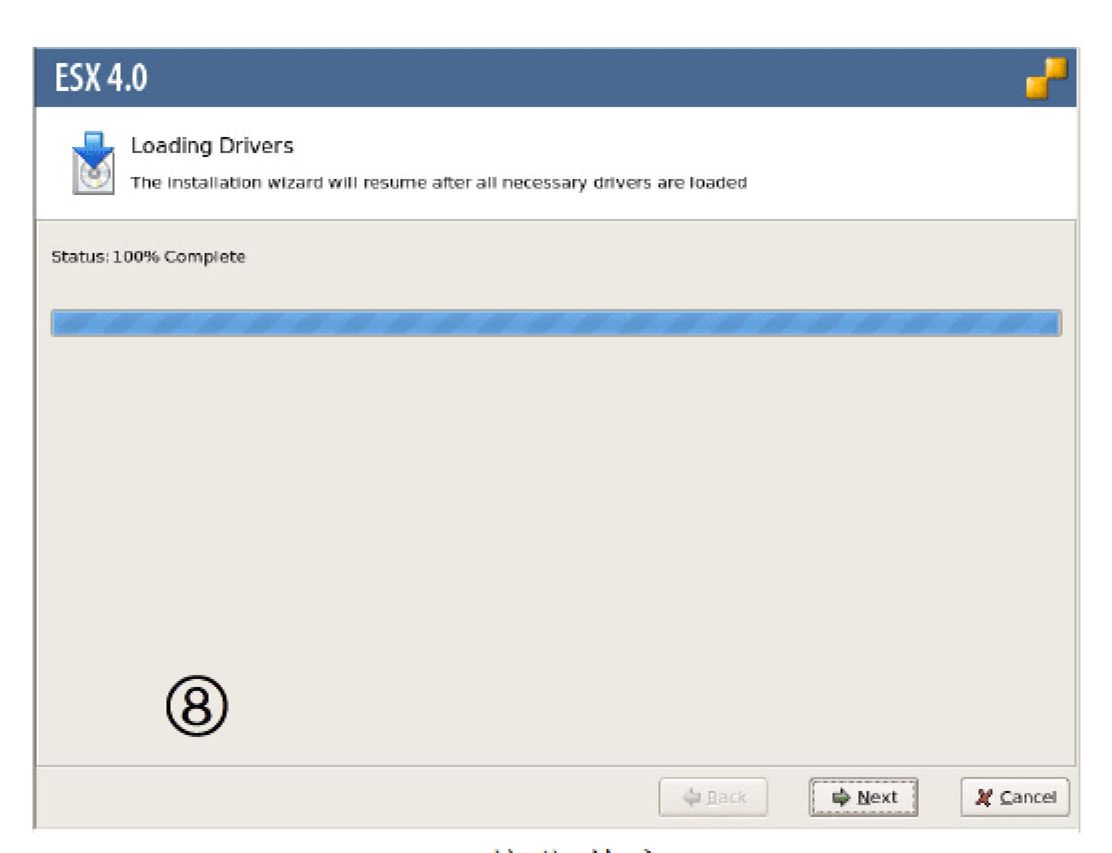
▲ 一般不需要安装额外的驱动程序

7．会询问是否要装载系统驱动程序，单击 Yes 按钮。

8．接着会装载系统驱动程序。装载完之后单击 Next 按钮。



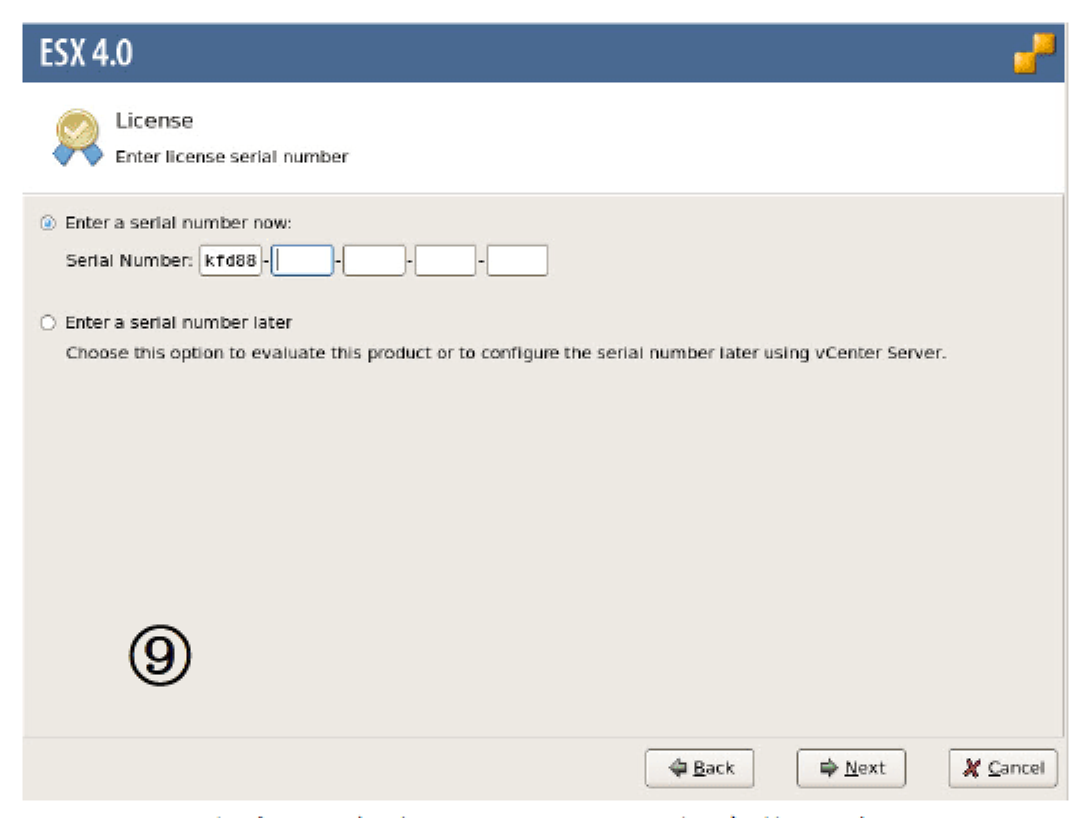
▲ 但系统要装载自身的驱动程序



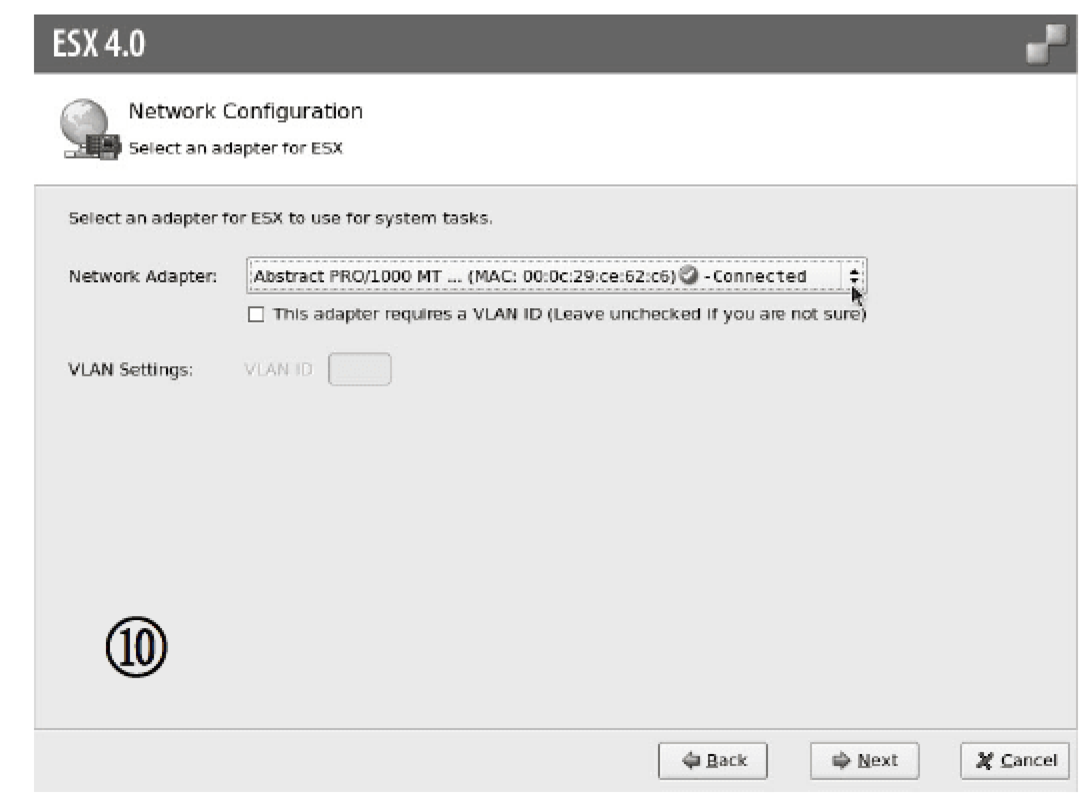
▲ 装载落实

9. 接着是键入产品序号，序号决定了 ESX 的功能。如果你只想试用的话，选择第二项，单击 Next 按钮。

10. 接下来是选择总控这台 ESX 的网络界面。如果有两片网卡，ESX 会默认绑定这两个网卡。



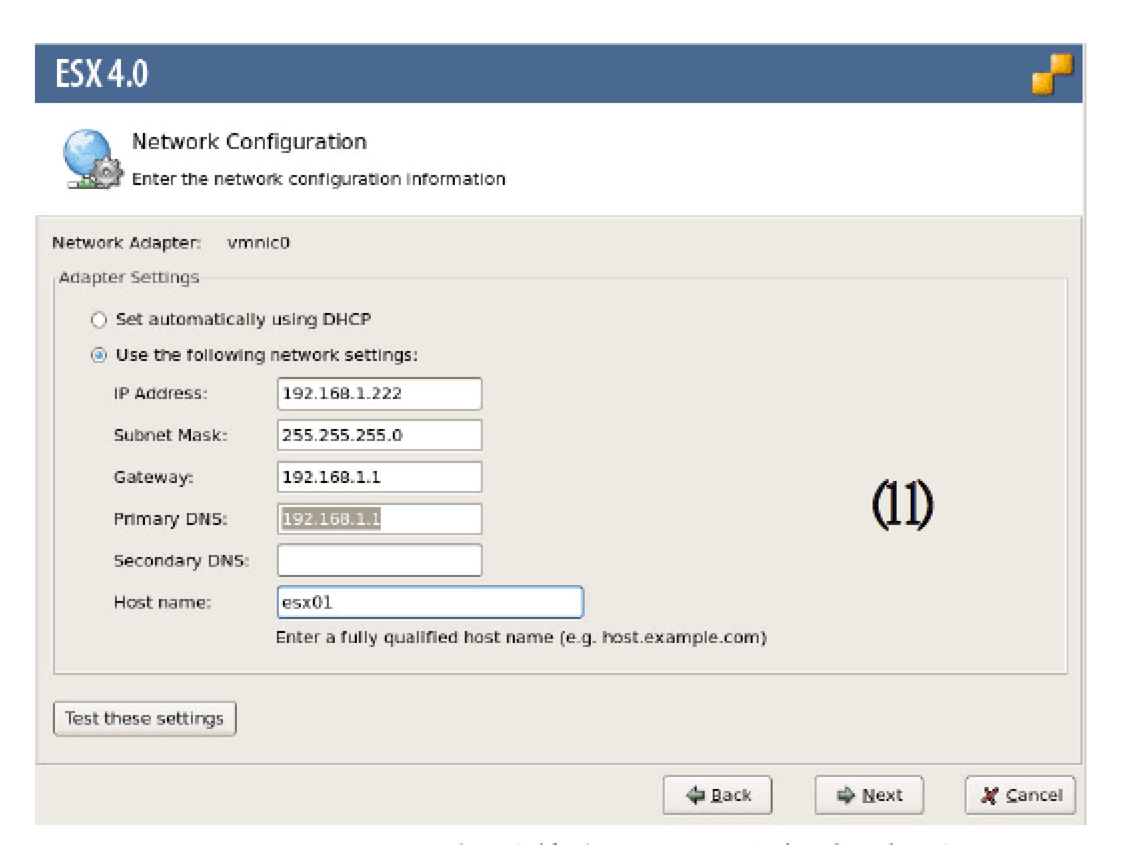
▲ ESX 的序号决定了 vSphere 的功能，如 VMotion 等



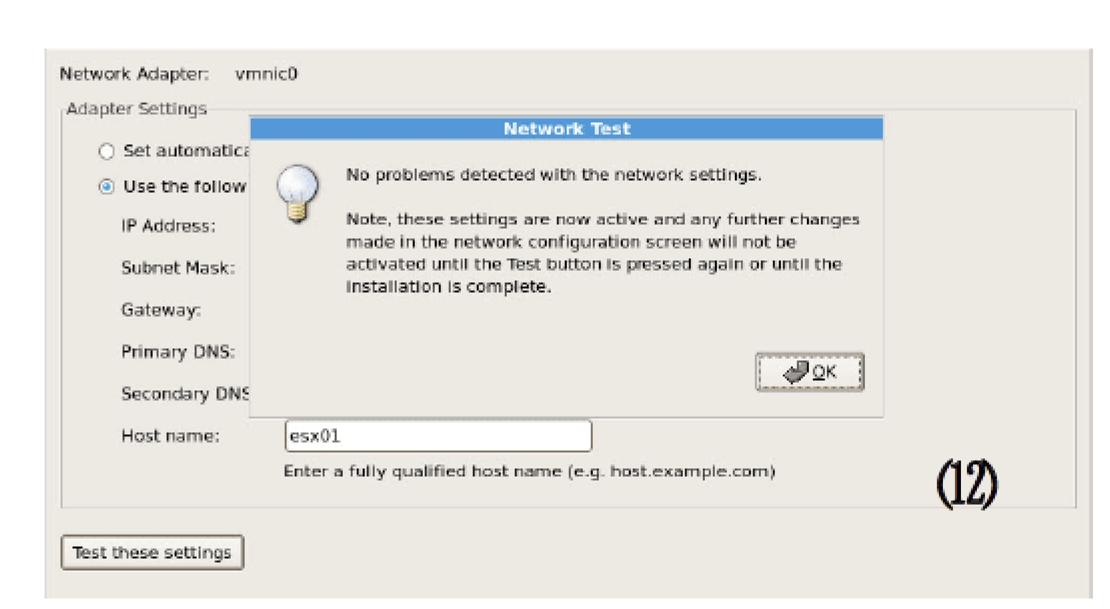
▲ 配置网络界面

11. 接下来是配置总控网络界面，可以使用 DHCP，但笔者不建议使用，还是自行配置 IP。我们会在 ESX 网络组态的章节详细说明 ESX 网络的重要性。

12. 配置完之后可以单击 Test these settings 按钮确定这个网络配置是否常规。如果弹出图中的画面，意味着网络配置常规。



▲ ESX/ESXi 的网络界面配置十分重要，在网络配置的部分会有详细说明



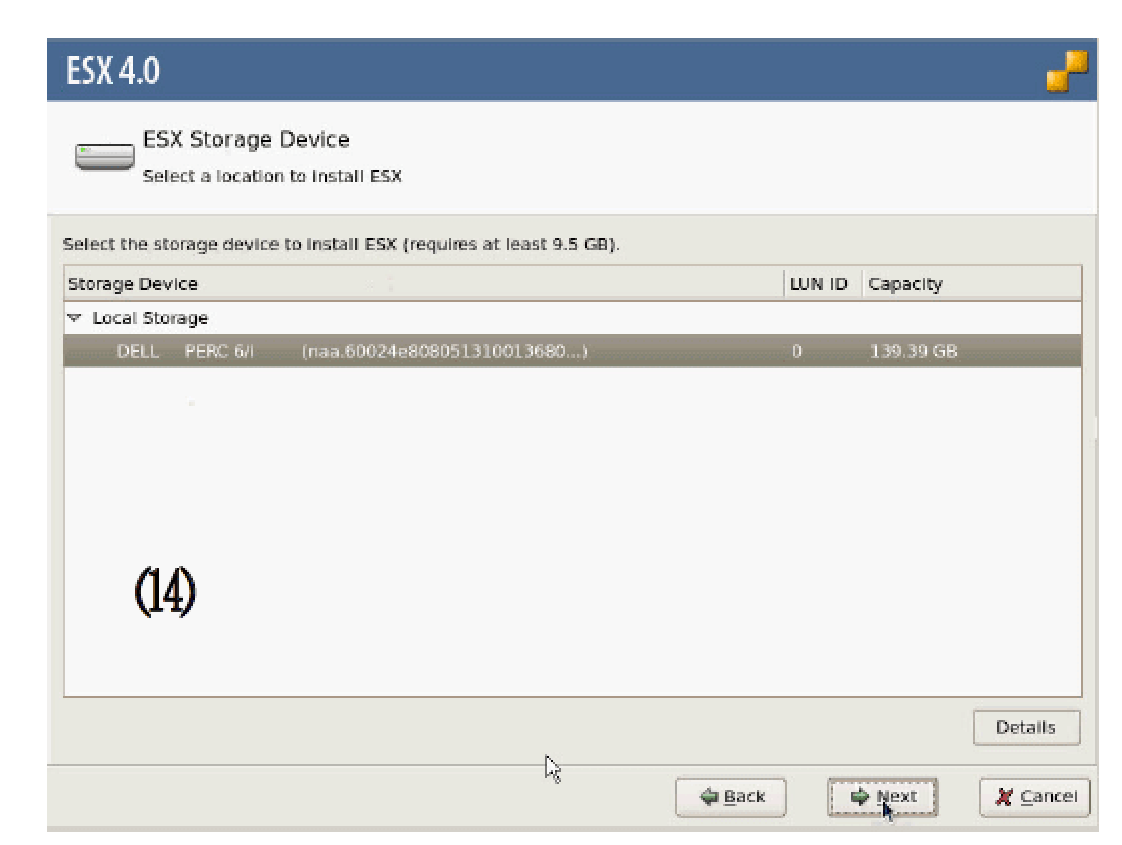
▲ 测试配置是否成功

13. 接下来会让你选择安装的方式，我们一般选择 Standard，系统会自动做硬盘的分割区。

14. 接下来会弹出可安装的硬盘，我们可以看到刚才安装的 Perc 6/i 的 SATA 上有一个做好的 RAID 硬盘。单击 Next 按钮。



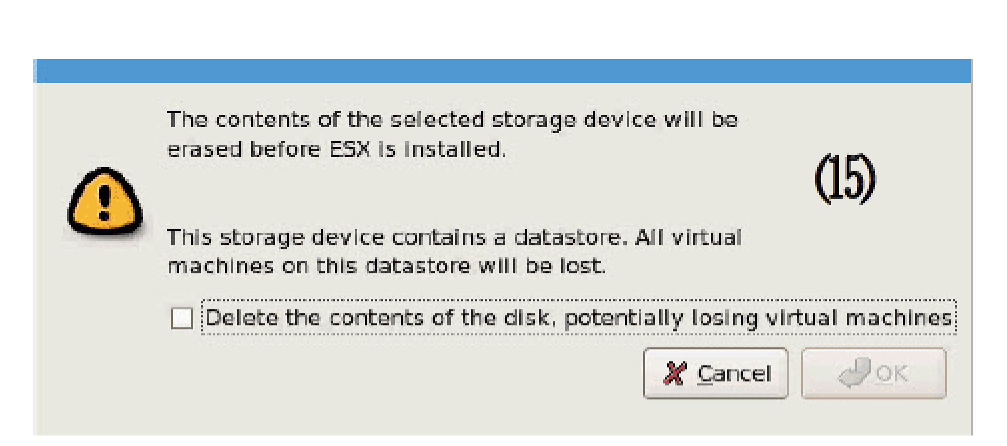
▲ 一般选择 Standard 让系统来分区



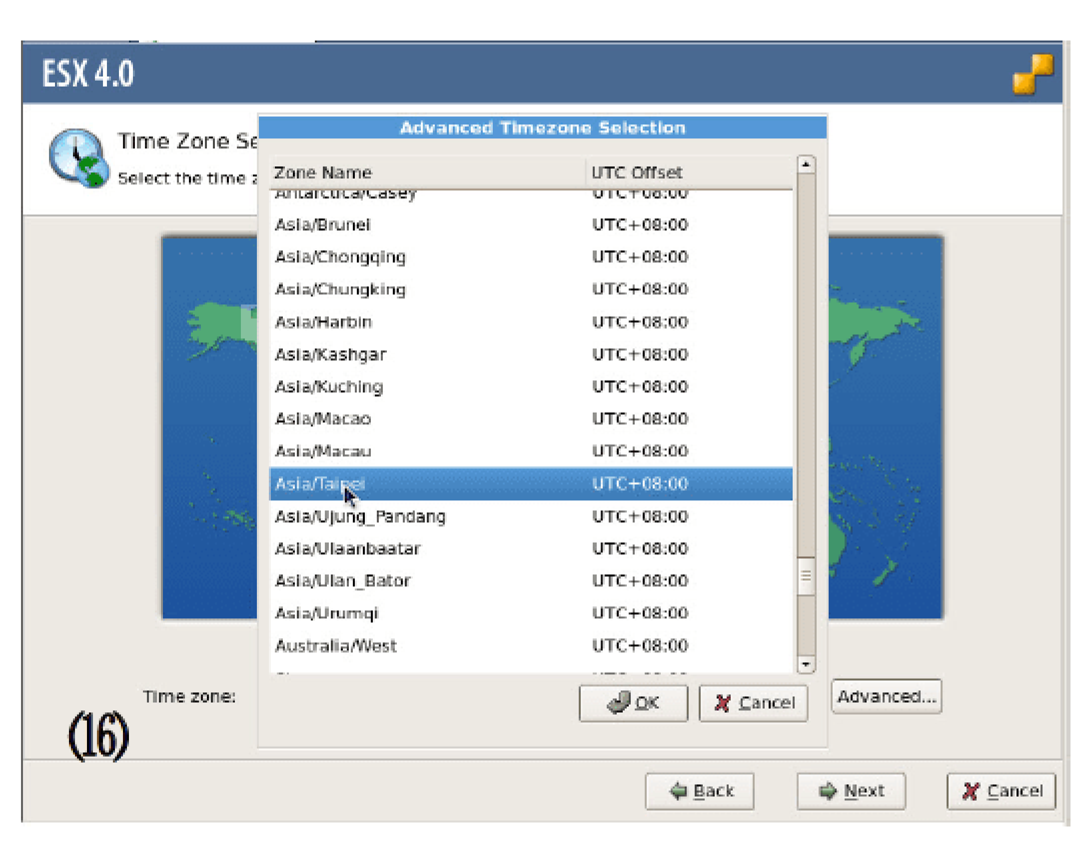
▲ 选择系统能抓到的硬盘

15. 此时系统会说明，这个硬盘上的所有分割区都会被清除 确定选中之后单击 OK 按钮。

16. 接下来是选择时区，选择你所在位置的时区即可，单击 Next 按钮。



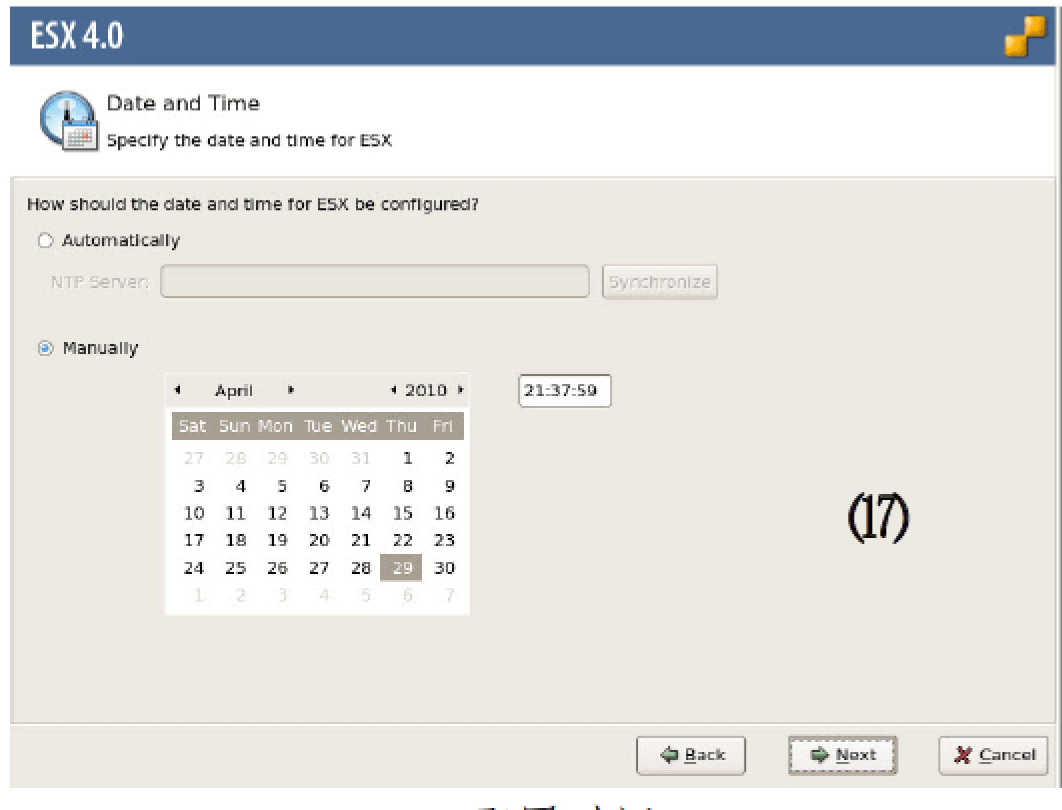
▲ 会确定删除硬盘数据



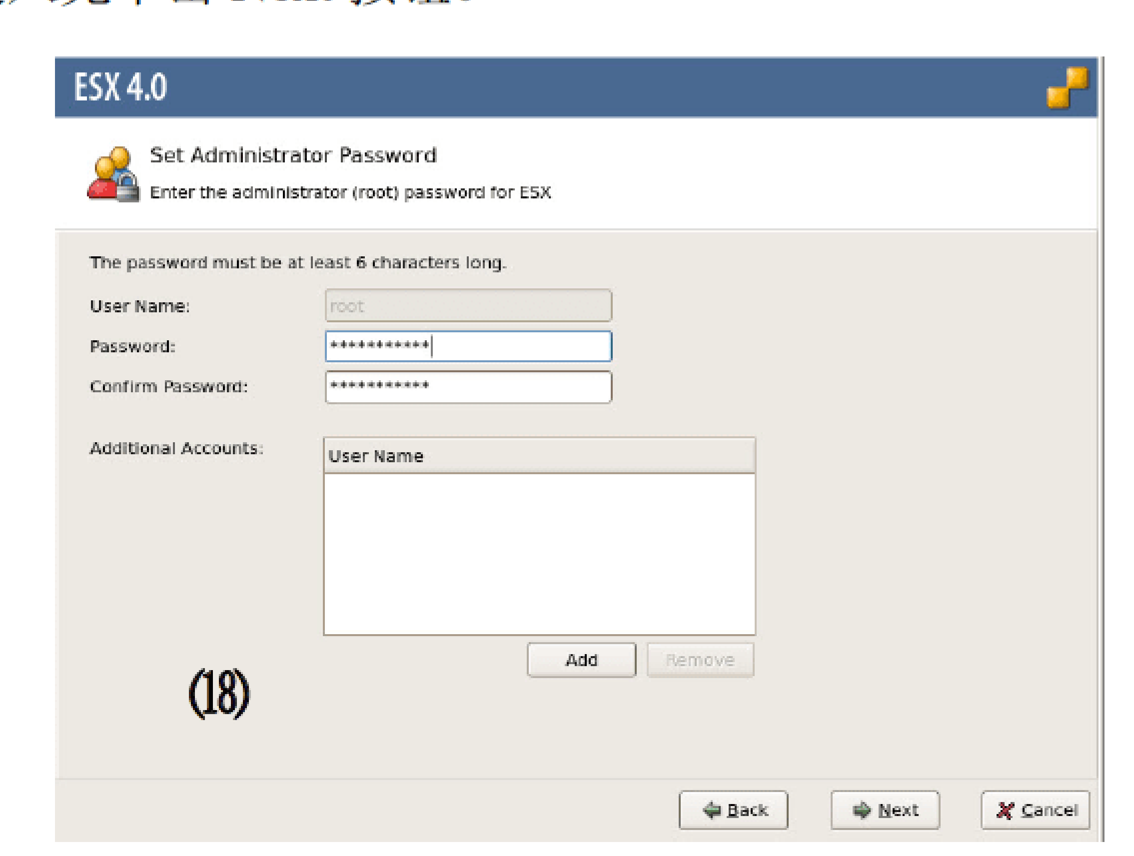
▲ 选择时区

17. 接下来是配置时间，我们选择手动，键入完单击 Next 按钮。

18. 接下来是配置 root 的账号口令，键入完单击 Next 按钮。

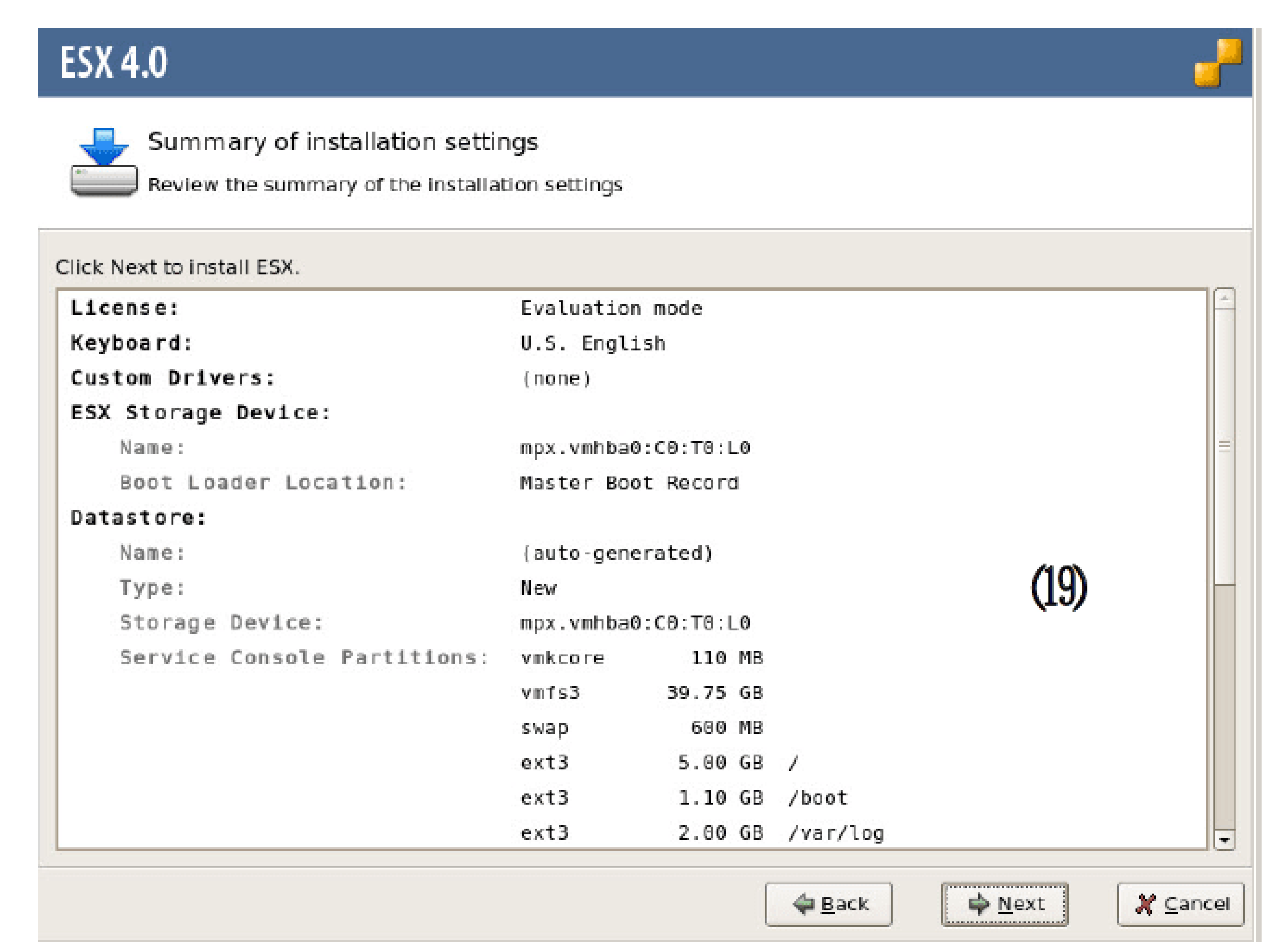


▲ 配置时间



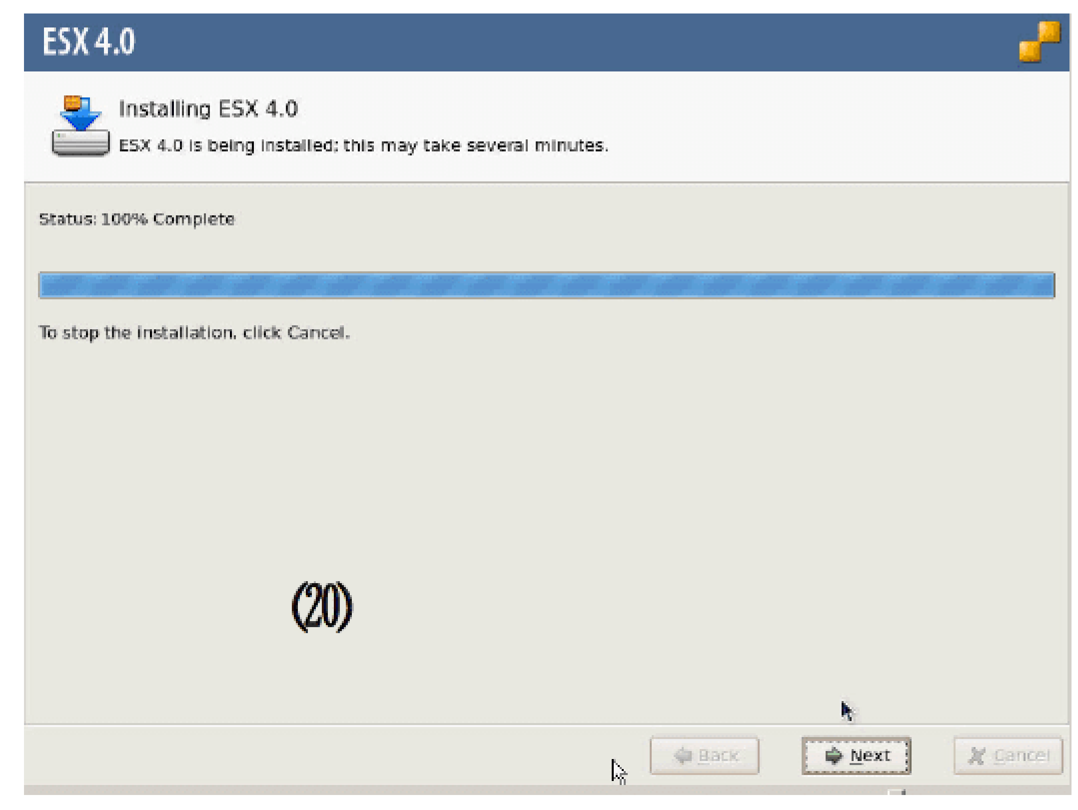
▲ ESX 的 Linux 通常是以 root 登录

19. 接下来是总结画面，确定之后单击 Next 按钮。

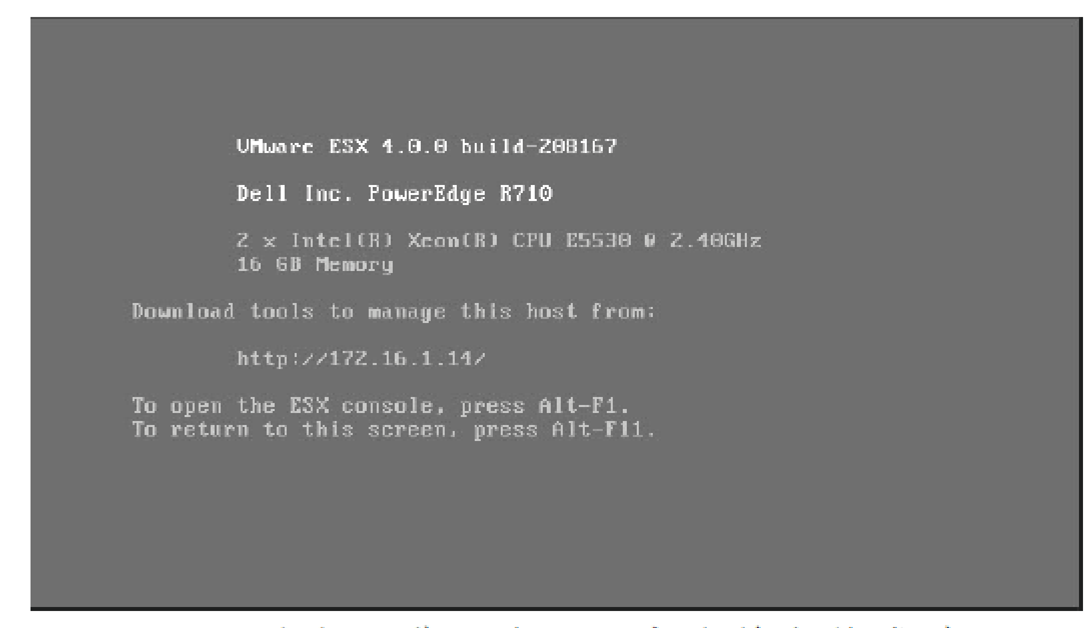


▲ 总结画面之前都可以回去修改配置

20. 接着系统会开始安装，当到 100%时，按下 Next。到此 ESX 已经安装落实，按下 Finish 后会重新激活计算机。



▲ 到此安装完毕



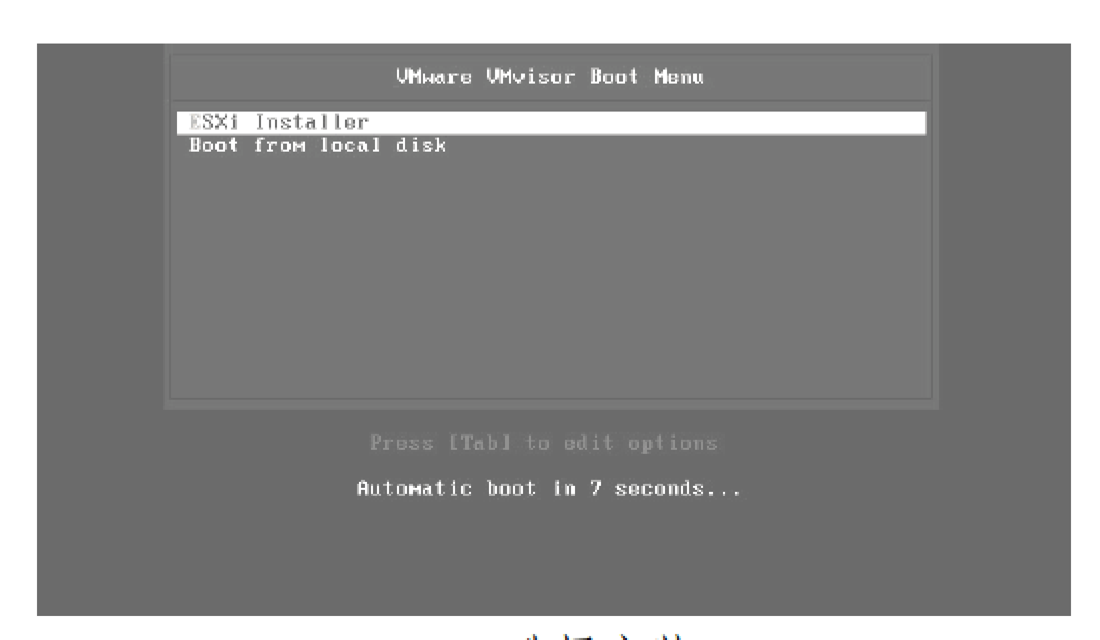
▲ 重启后进入此画面意味着安装成功

8.2.2 在物理机上安装 ESXi

只要准备好硬件环境，安装 ESXi 也十分简单，我们就在这一小节进行说明。

▶ 安装 ESXi

1. 将光盘放入服务器的光盘机，或是使用 Virtual Device 挂入 ISO 档，参阅前一节的安装过程。
2. 当 ESXi 引导后会弹出如图的画面，选择 ESXi Installer 选项。
3. 此时会装载驱动程序，时间会稍久。



▲ 选择安装

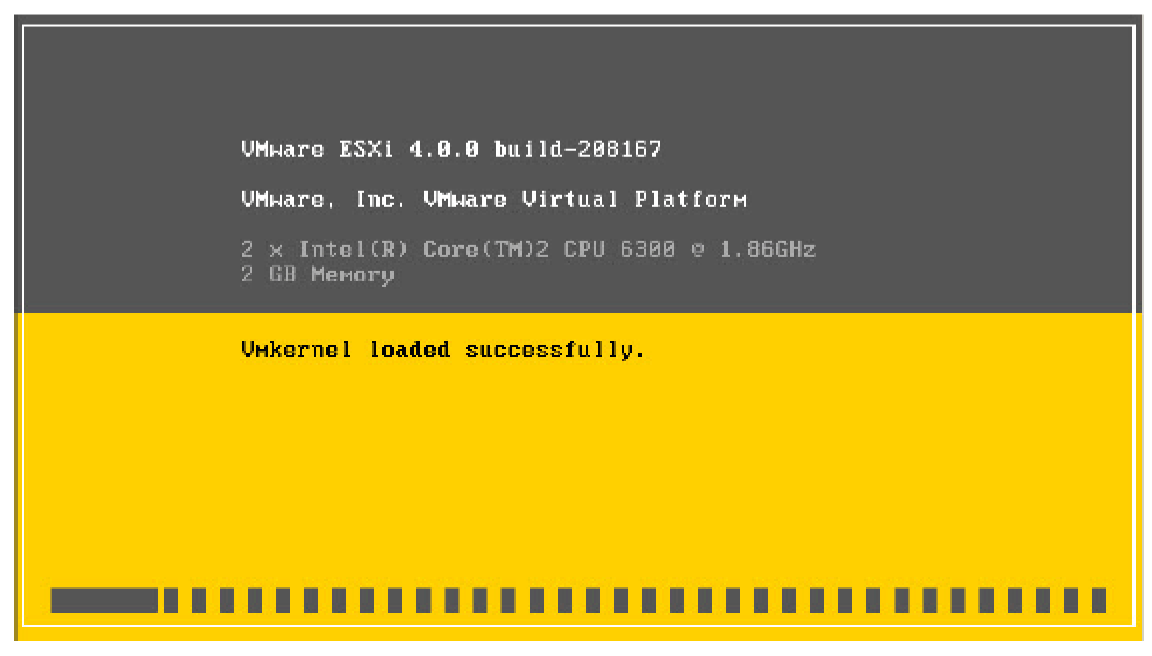
```
Press [Tab] to edit options

Automatic boot in 1 second...

COM32 Multiboot loader v0.2.  Copyright (C) 2005-2006 Tim Deegan.
Kernel: vmkboot.gz
Loading vmkboot.gz..
Module: vmkernel.gz
Loading vmkernel.gz......
Module: sys.vgz
Loading sys.vgz......
```

▲ 装载 ESXi 的时间会较久

4．接下来会装载标准的 ESXi 驱动程序画面，和 ESX 很像。



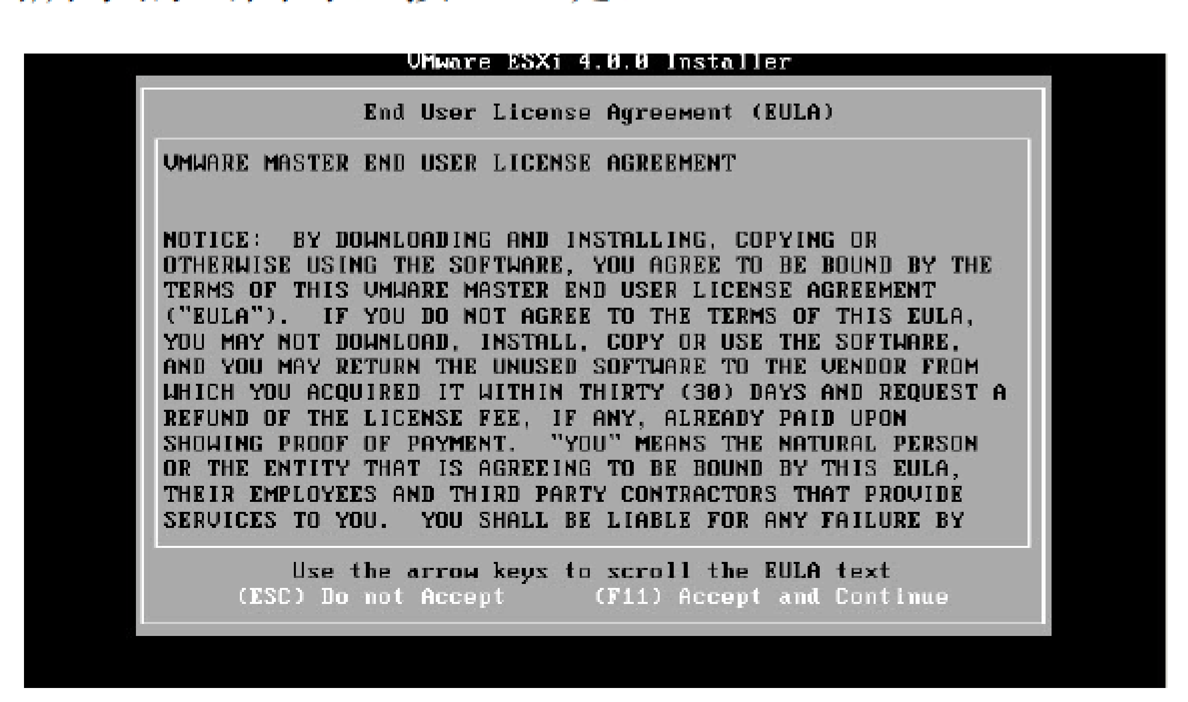
▲ 标准的驱动程序装载画面

5．弹出如下图所示的安装画面，按 Enter 键。



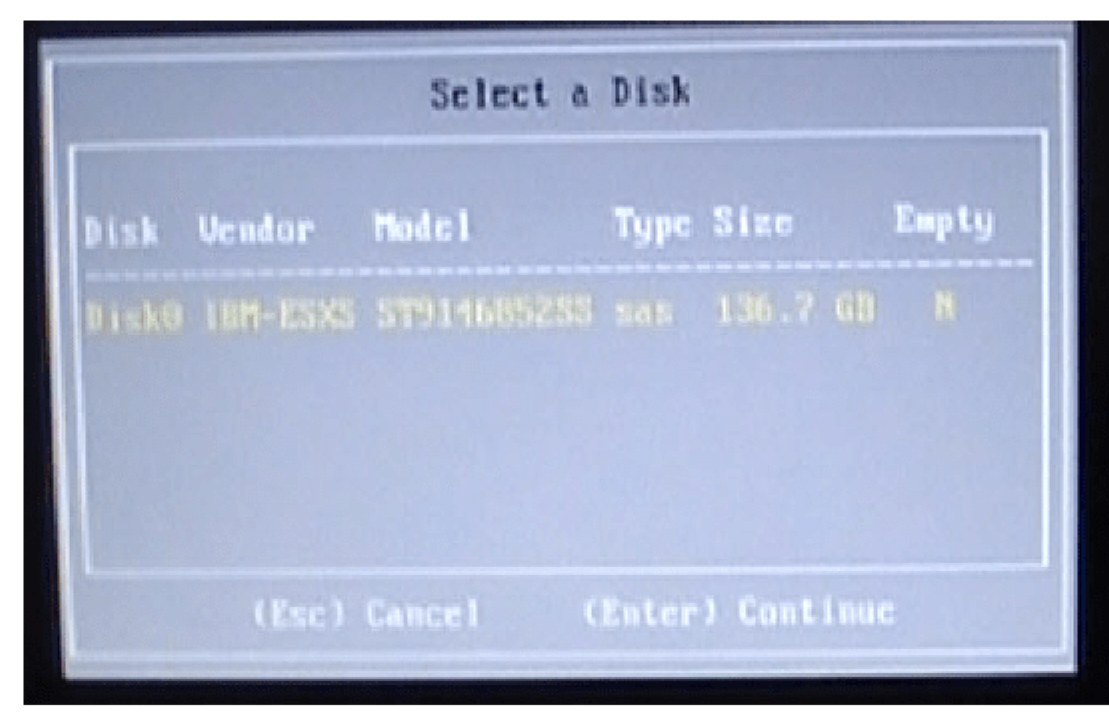
▲ 按 Enter 键

6．弹出如下图所示的画面时，按 F11 键。



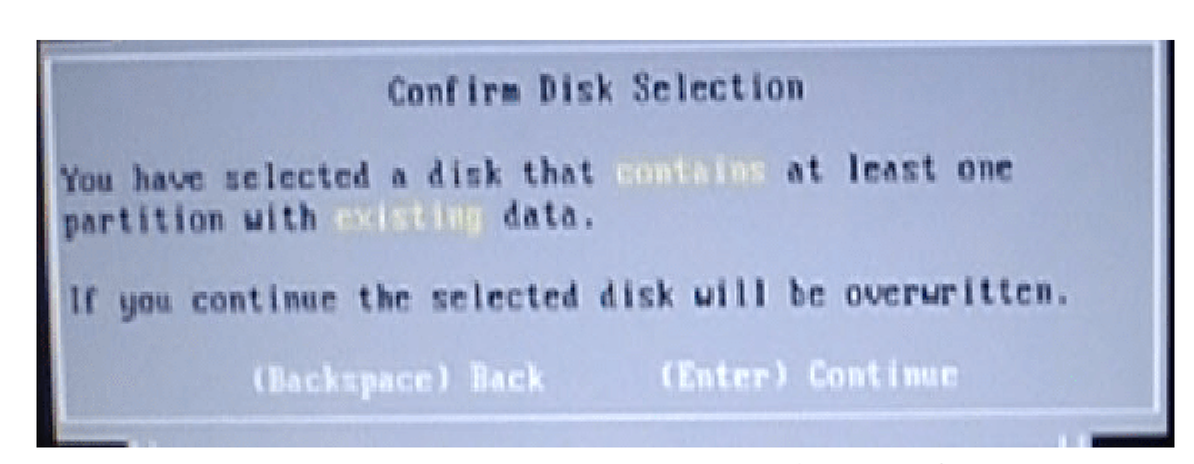
▲ 在 ESXi 中大部分都是按 F11 键做决定

7．接下来是硬盘安装画面。会弹出系统可用的硬盘，按 Enter 键继续。



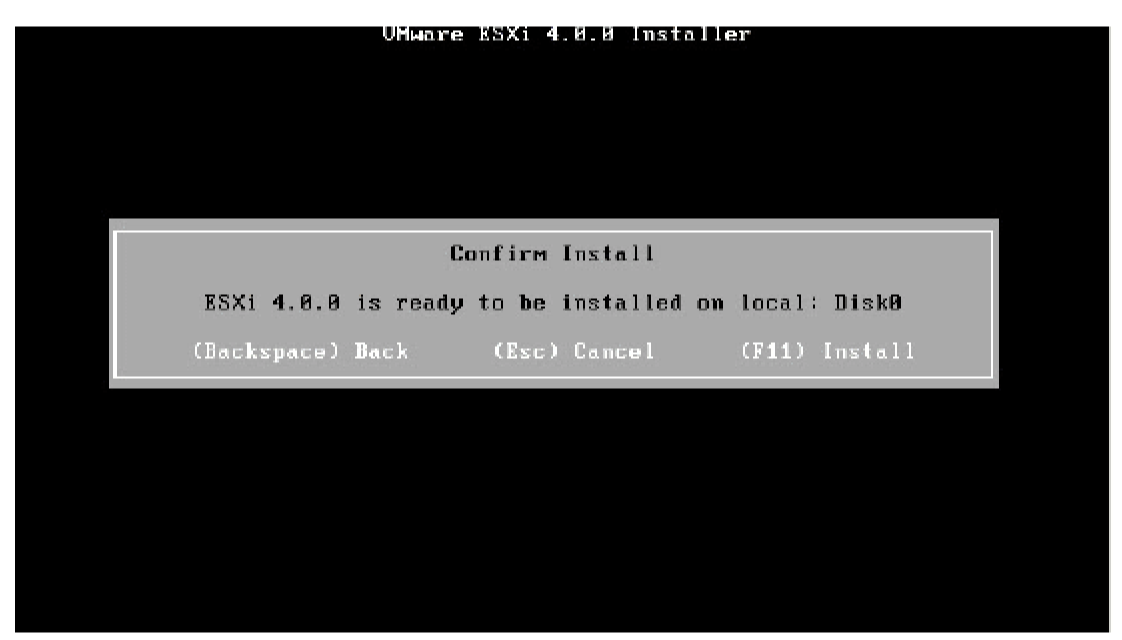
▲ 会给出可使用的硬盘

8．如果硬盘中已经有数据，会弹出确定删除的画面，按 Enter 键继续。



▲ 如果硬盘有数据也会确定删除

9．此时会弹出确定安装的画面，按 F11 键继续。



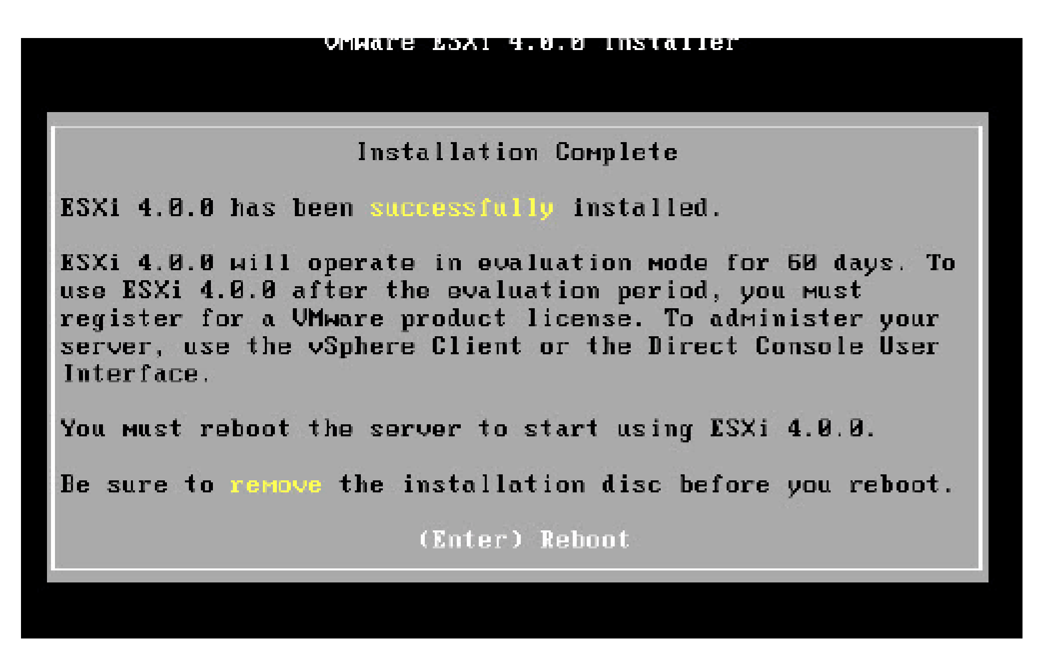
▲ 按 F11 键继续

10．接下来会开始安装，当到 100%时即安装完毕。



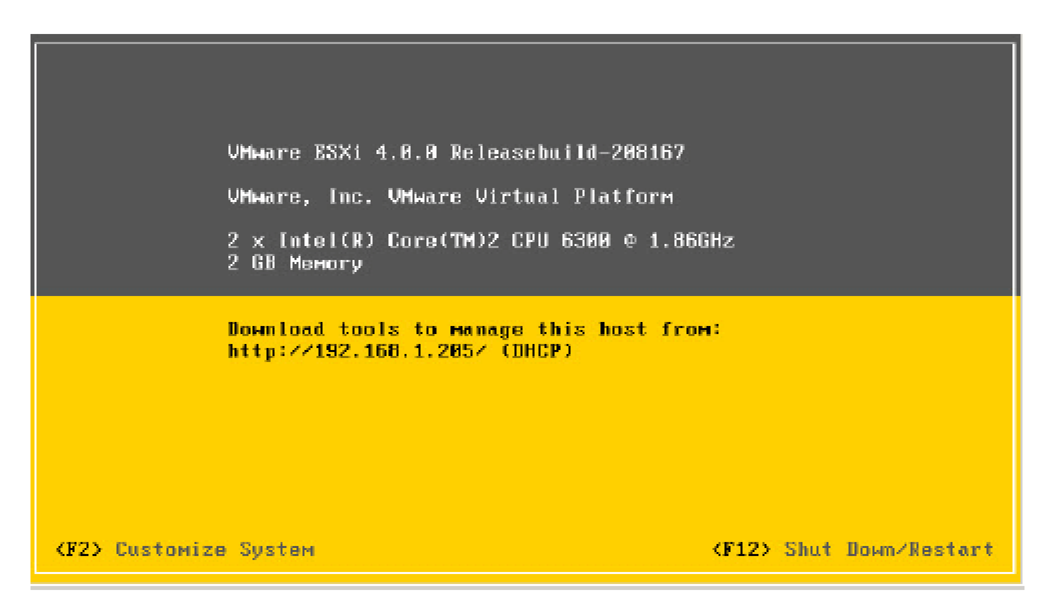
▲ 到 100%即安装完毕

11．当弹出图中的画面时，意味着已安装落实，按 Enter 键重新激活，系统会有提示。



▲ 安装落实重新激活

12．当弹出如下图所示的画面时，意味着已经重启进入 ESXi 中。



▲ 成功引导进入 ESXi 了

8.2.3 在 VMware Workstation 上安装 ESX/ESXi

ESX 必须在匹配 VMware HCL 的硬件上才能安装，因此一般的 PC 要安装上必须靠运气，但如果只是想评估 ESX，难道一定要买一台匹配 HCL 的服务器吗？事实上，VMware 为了让每一个人都可以体会 ESX 的强大，特别在新版的 VMware Workstation 7.1 上激活了对 ESX 的支持，让你可以在“虚拟”的环境中再安装 Hypervisor，并且也可以在上面再安装 VM。虽然效能无法应用在正规环境，但至少可以评估 ESX 的各项功能，如 HA、VMotion、DRS 等，下面就来看看完整的安装步骤。

1. 在 Workstation 上安装 ESX/ESXi 的必要条件

虽然大部分的个人计算机都可以安装 VMware Workstation，但 ESX/ESXi 是一个 64 比特的 Hypervisor，因此运行 VMware Workstation 的机器必须拥有 64 比特的 CPU，而 CPU 也必须支持 VT-X 技术，满足这两个条件才能在上面安装 ESX/ESXi。

2. 创建给 ESX/ESXi 用的 VM

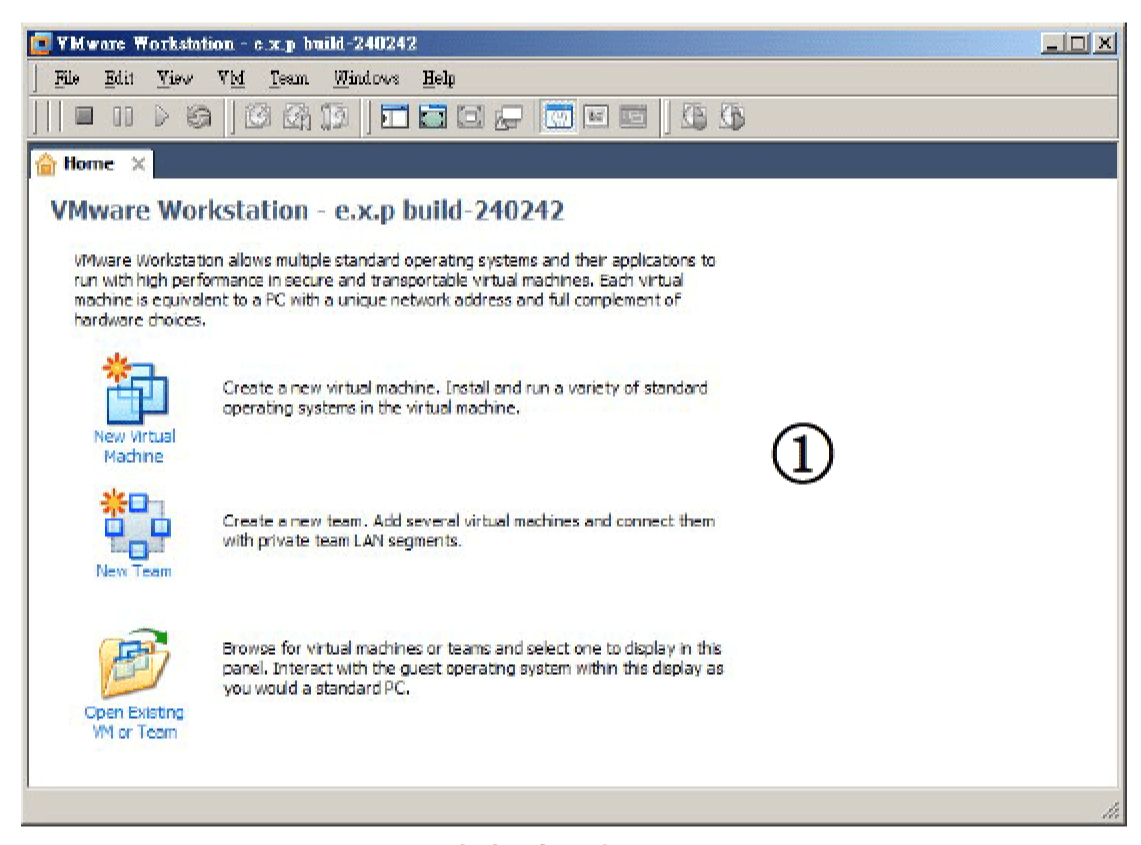
VMware Workstation 很贴心地在其上特别准备了给 ESX/ESXi 运行的环境，只要经过简单的配置步骤即可。

3. 创建给 ESX/ESXi 用的 VM

在 VMware Workstation 下创建给 ESX/ESXi 的 VM 和创建其他的 VM 一样简单。

1．进入 VMware Workstation 并且选择 New Virtual Machine 选项。

2．选择 Custom 单选按钮，单击 Next 按钮。

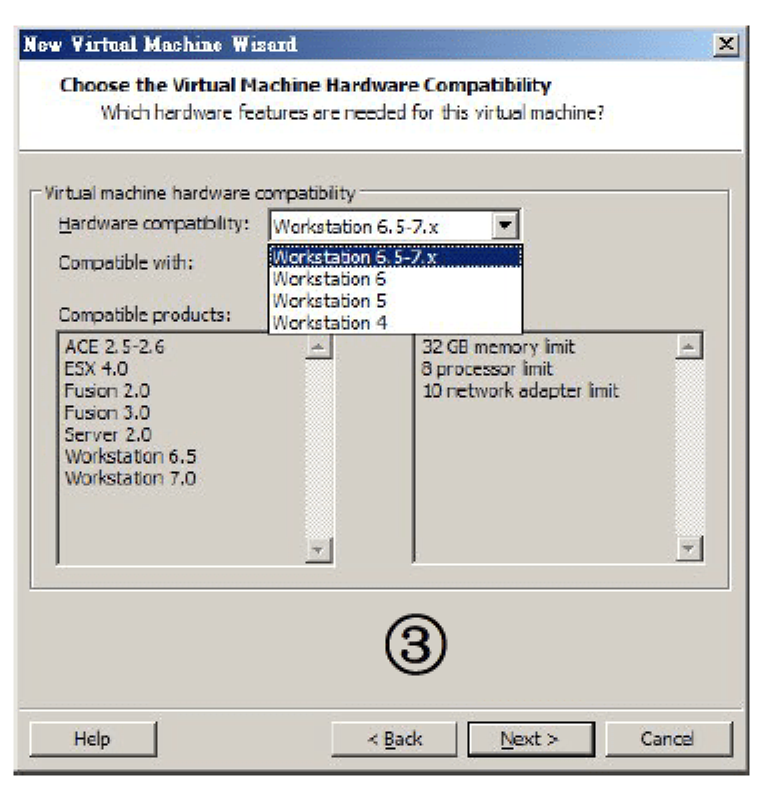


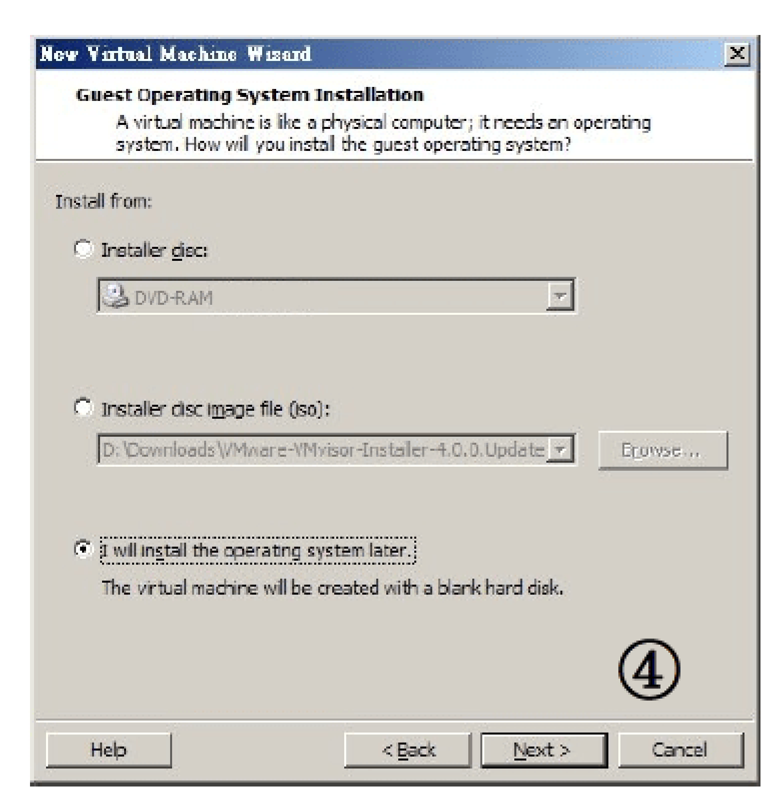


▲ 选择新建 VM　　▲ 选择第二项

3．选择 VMware Workstation 的版本，我们就选择默认值即可，单击 Next 按钮。

4．选择最后一个选项自行安装，单击 Next 按钮。

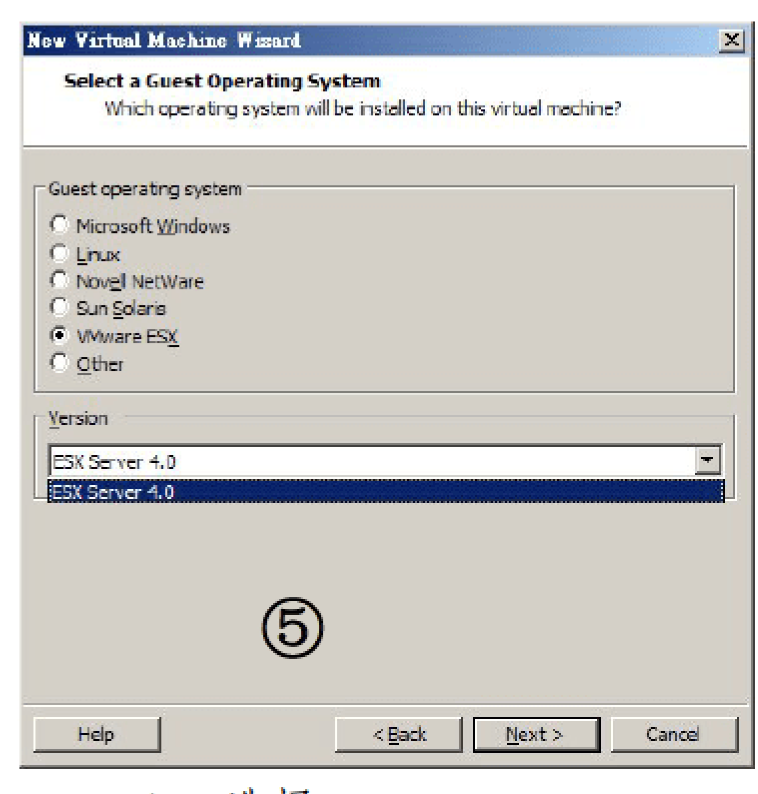


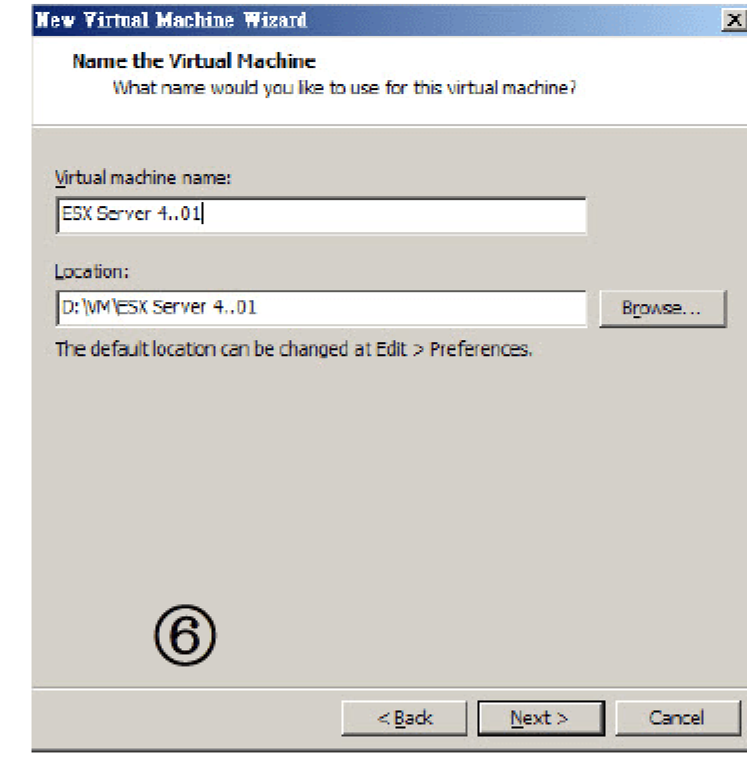


▲ 选择 Workstation 的版本，我们就按默认值即可　　▲ 选择最后一项即可

5．在操作系统的选择上，先选择 VMware ESX 单选按钮，在 Version 列表框选择 ESX Server 4.0。单击 Next 按钮。

6．在这里键入这个 VM 的名称，单击 Next 按钮。

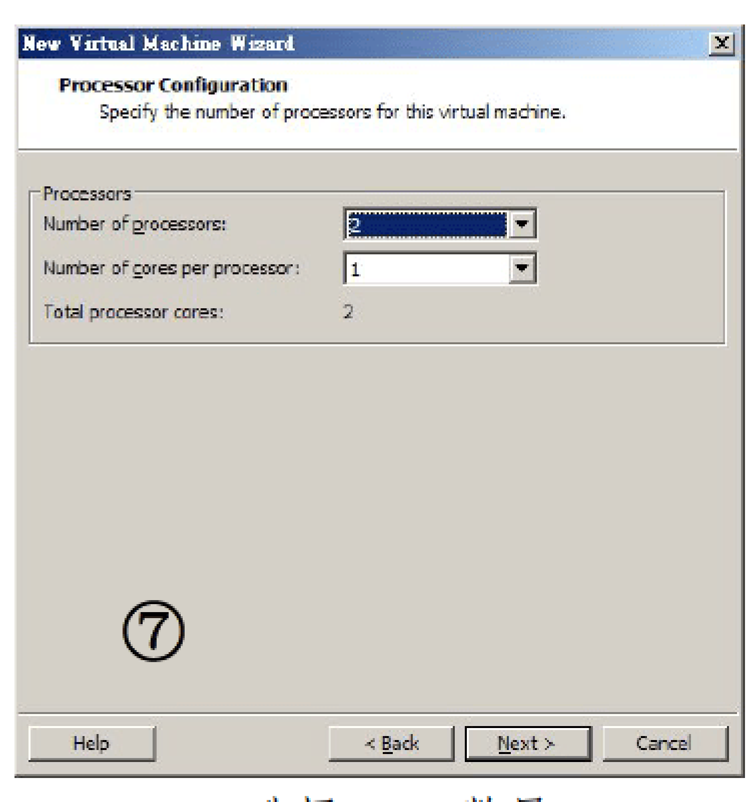




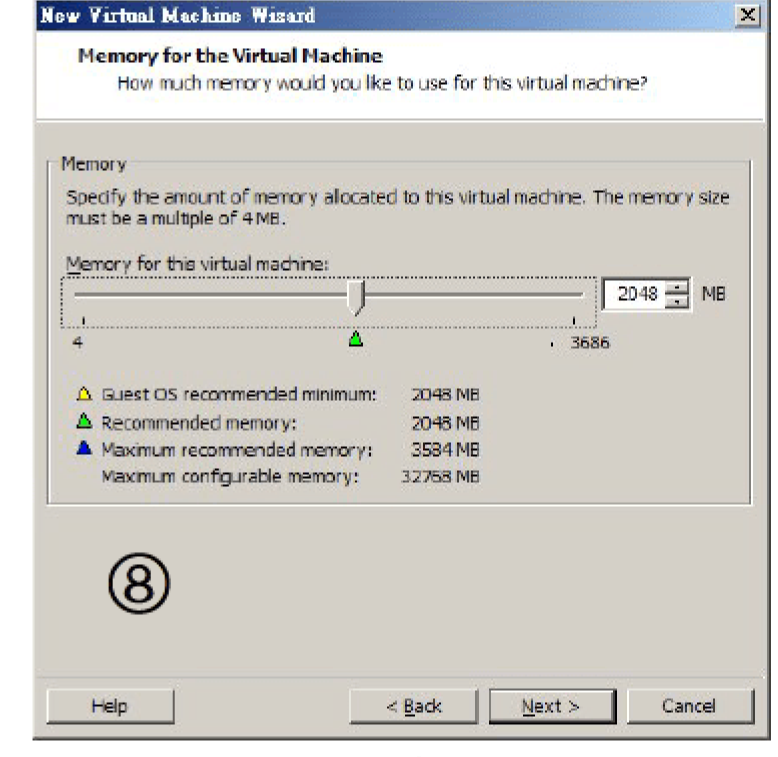
▲ 选择 ESX Server 4.0　　▲ 键入 VM 的名称

7．接下来选择 CPU 数量，默认值是 2CPU，单核，共 2 个内核，单击 Next 按钮。

8．接下来是内存的大小，默认值是 2GB，单击 Next 按钮。



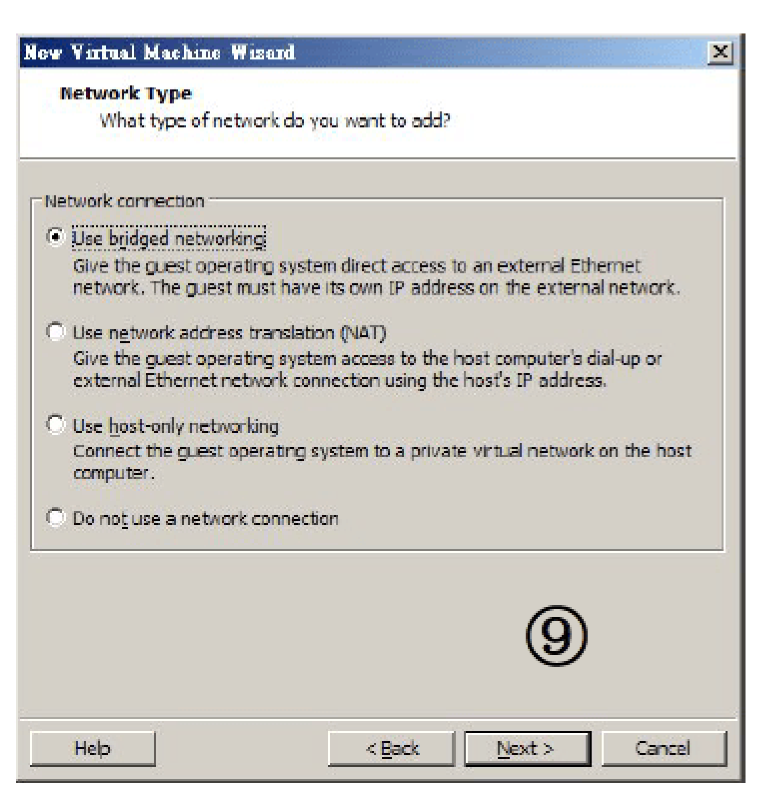
▲ 选择 CPU 数量



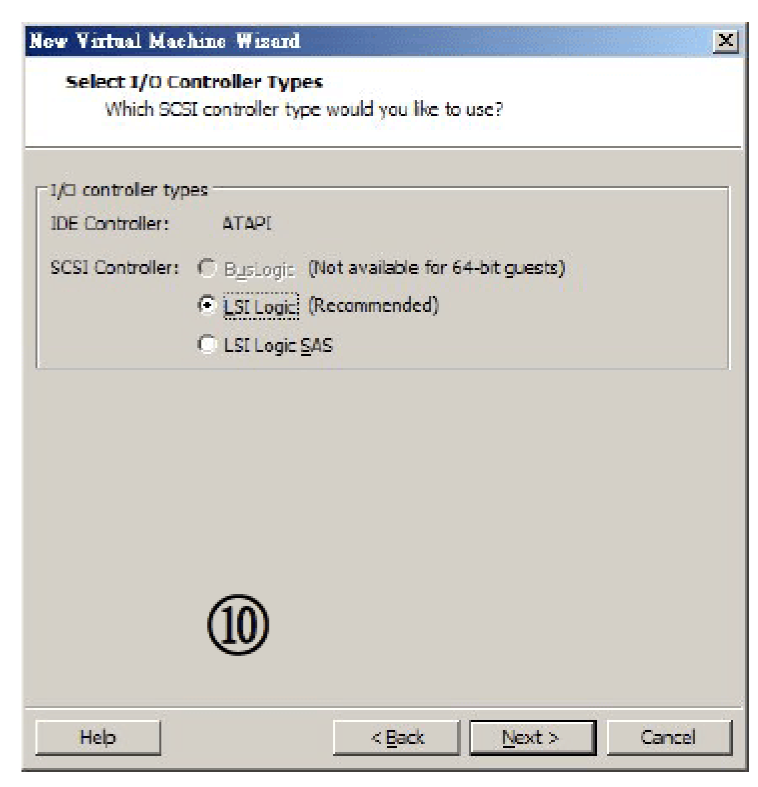
▲ 选择内存的大小

9．接下来是网络的选项，我们选择 Bridge 方便管理，单击 Next 按钮。

10．接下来是硬盘的界面卡，就以默认值为主，单击 Next 按钮。



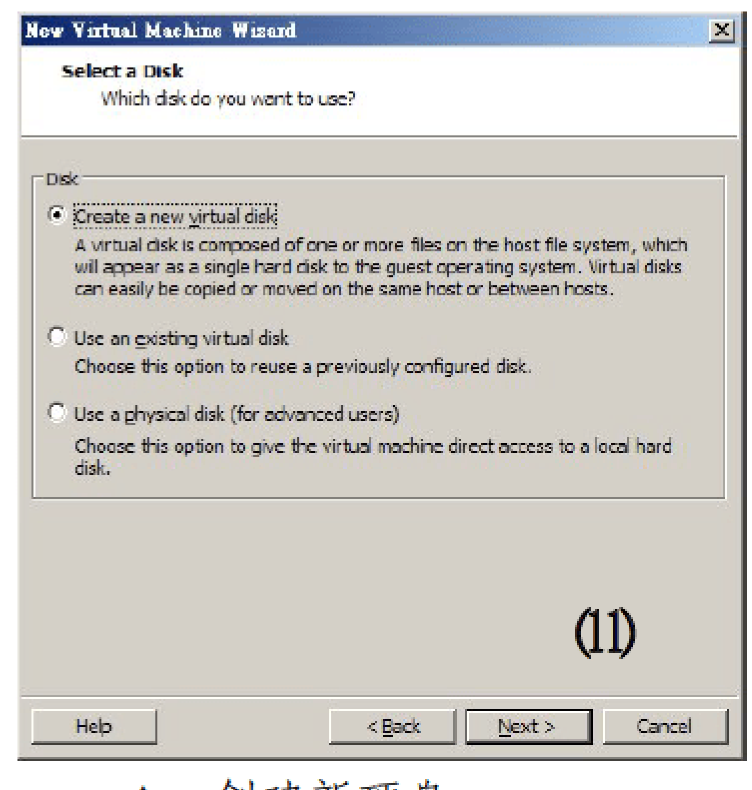
▲ 选择第一项 Bridged 可以在内网直接管理



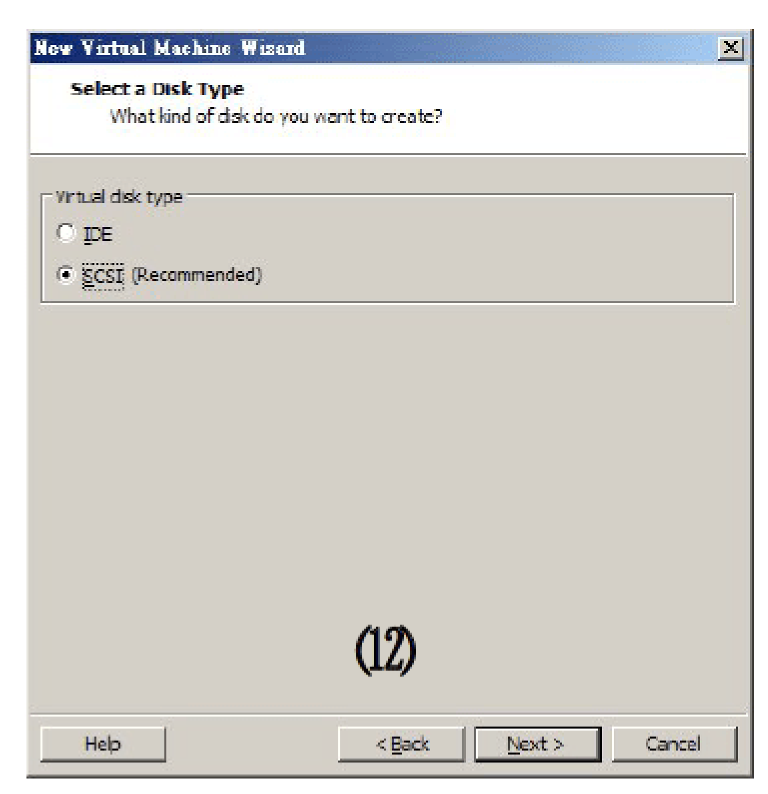
▲ 选择硬盘界面卡，以默认值即可

11．接下来是创建这个 ESX VM 的硬盘，选择第一个选项创建新硬盘，单击 Next 按钮。

12．接下来是硬盘的界面，ESX 以 SCSI 为主，单击 Next 按钮。



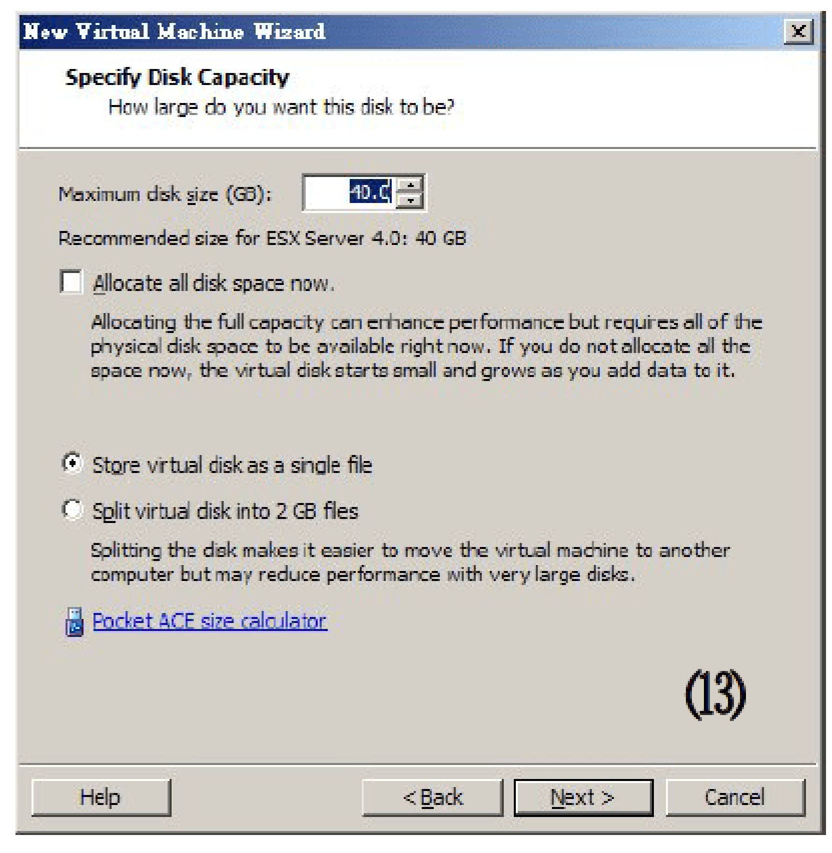
▲ 创建新硬盘



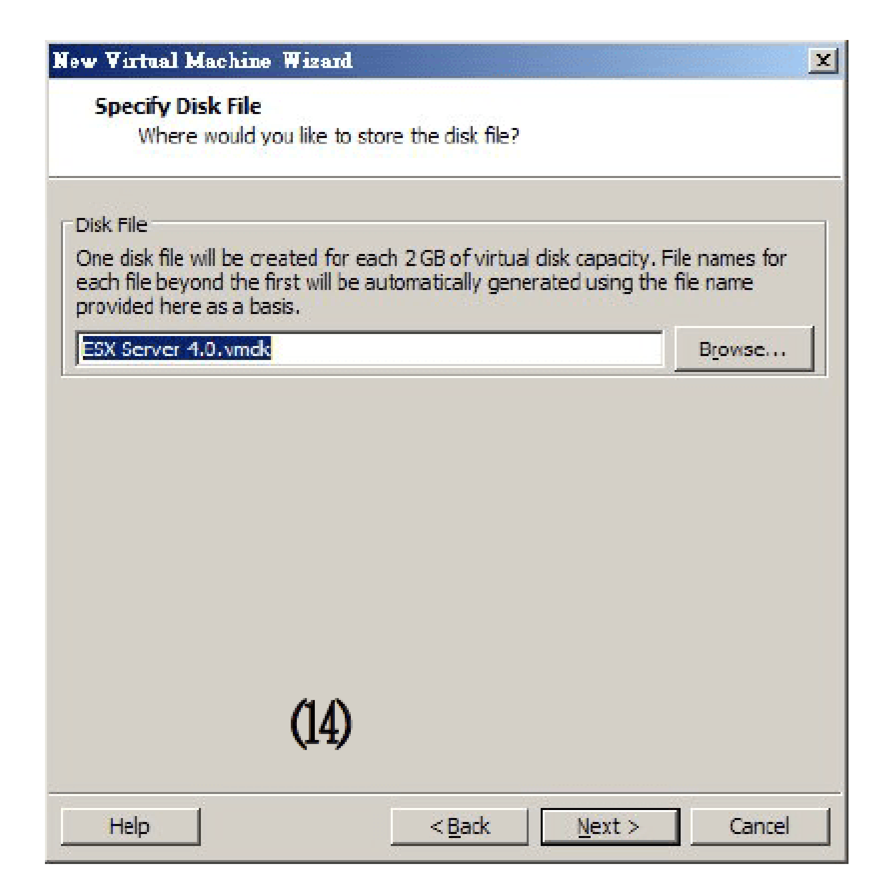
▲ SCSI 的兼容性较高

13．给定 VM 硬盘的大小，默认为 40GB，单击 Next 按钮。

14．接下来是这个 VM 使用硬盘 VMDK 的名称，也使用默认值，单击 Next 按钮。



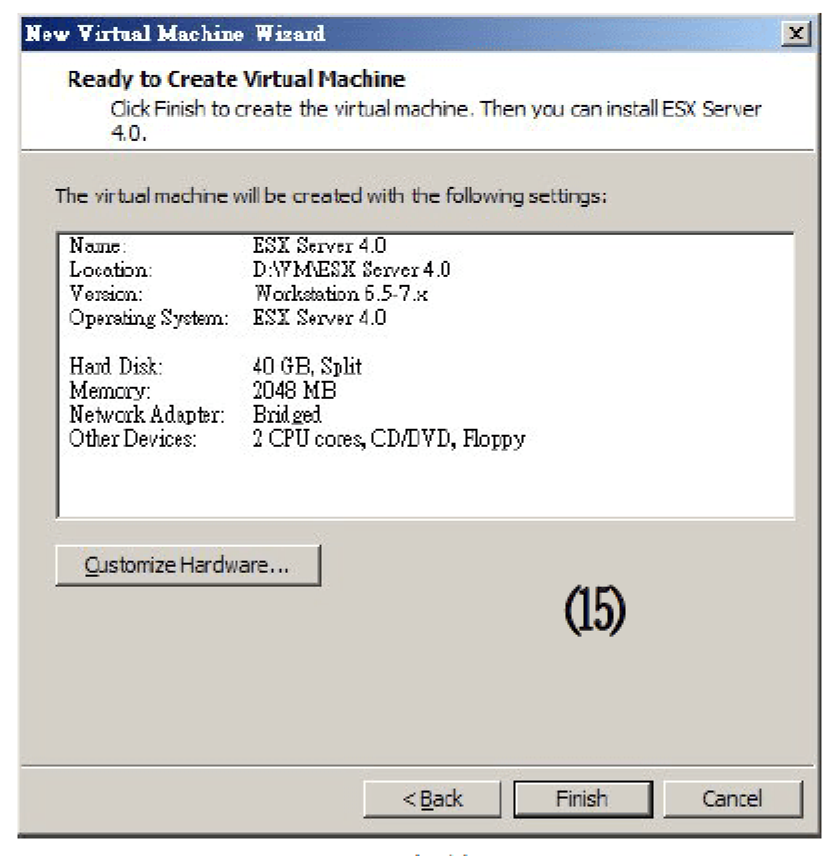
▲ 定义大小



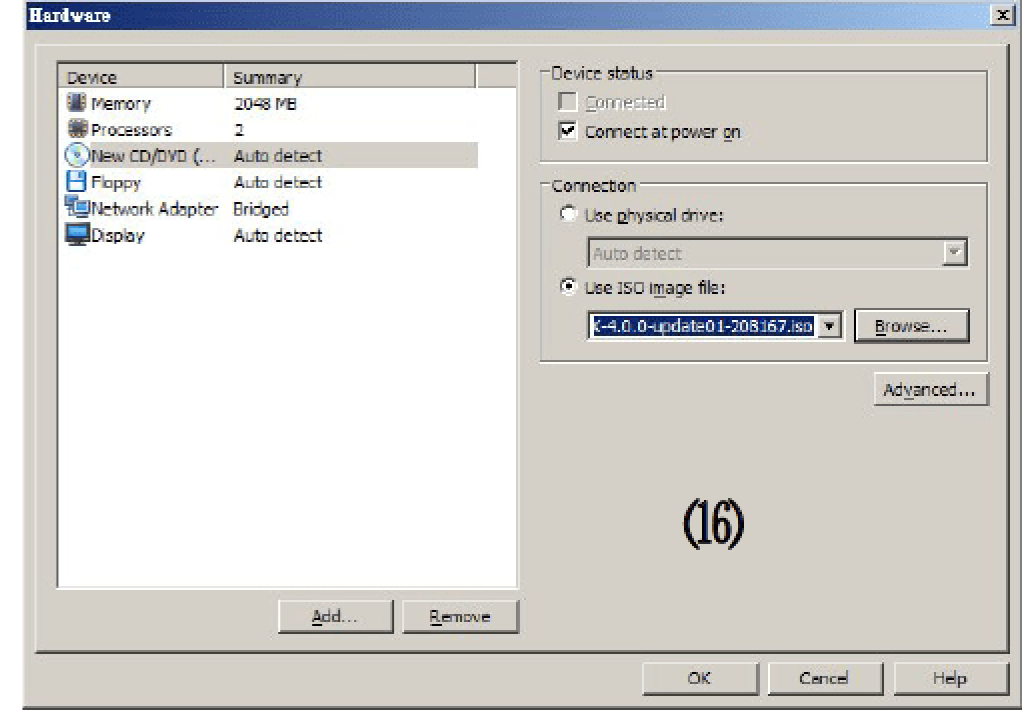
▲ 键入 VMDK 的文件名

15．接下来会弹出总结画面，单击 Customize Hardware 按钮。

16．我们将 CD/DVD 改成 Use ISO image file，可以将 ESX 光盘的 ISO 档挂入，选中引导时连接。单击 OK 按钮。

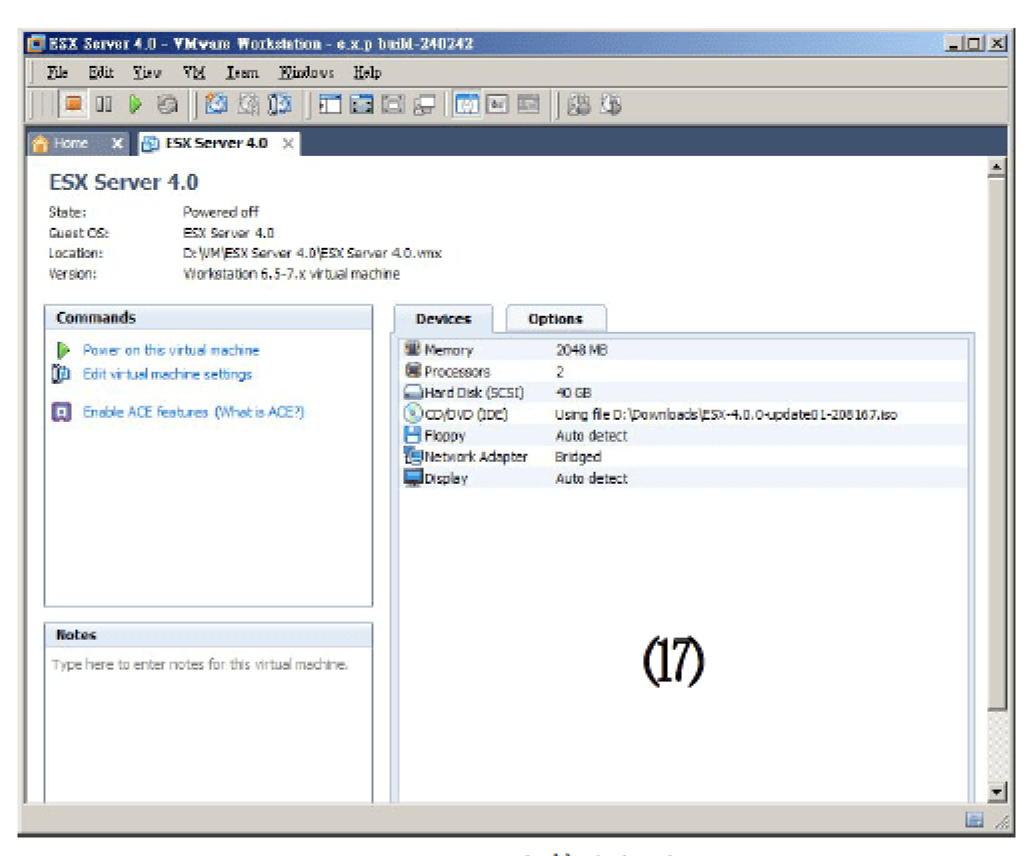


▲ 总结画面



▲ 挂入 ISO 档来引导

17．之后在步骤 15 的画面中，单击 Finish 按钮即可。此时我们会看到这个给 ESX 创建的 VM 已经可以使用了。

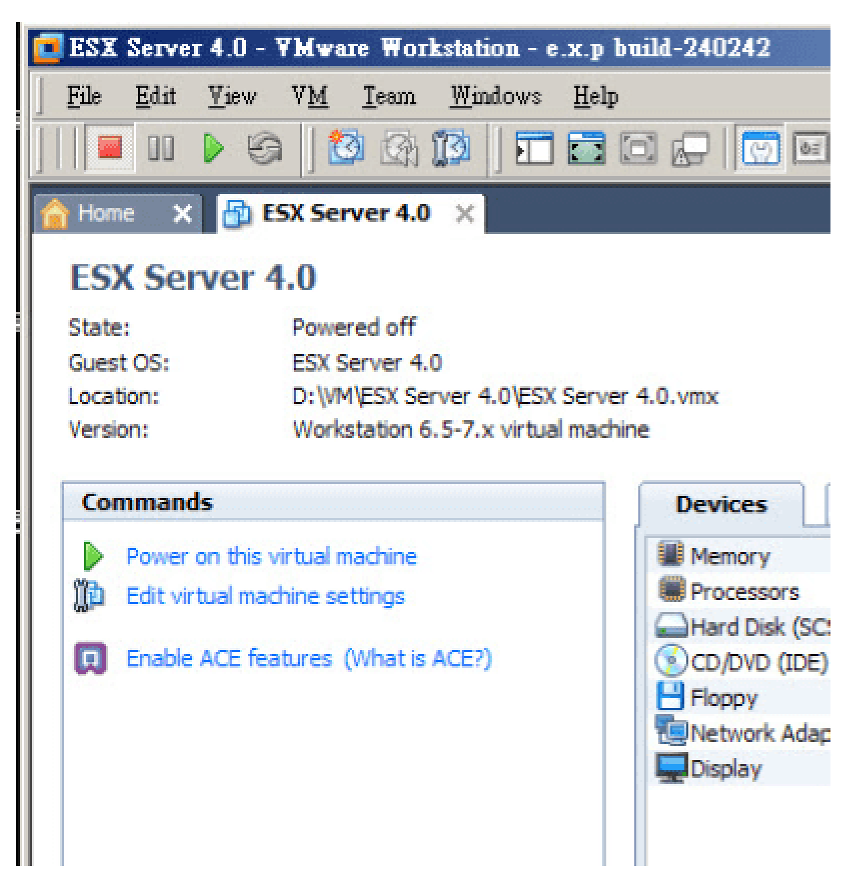


▲ 可以使用了

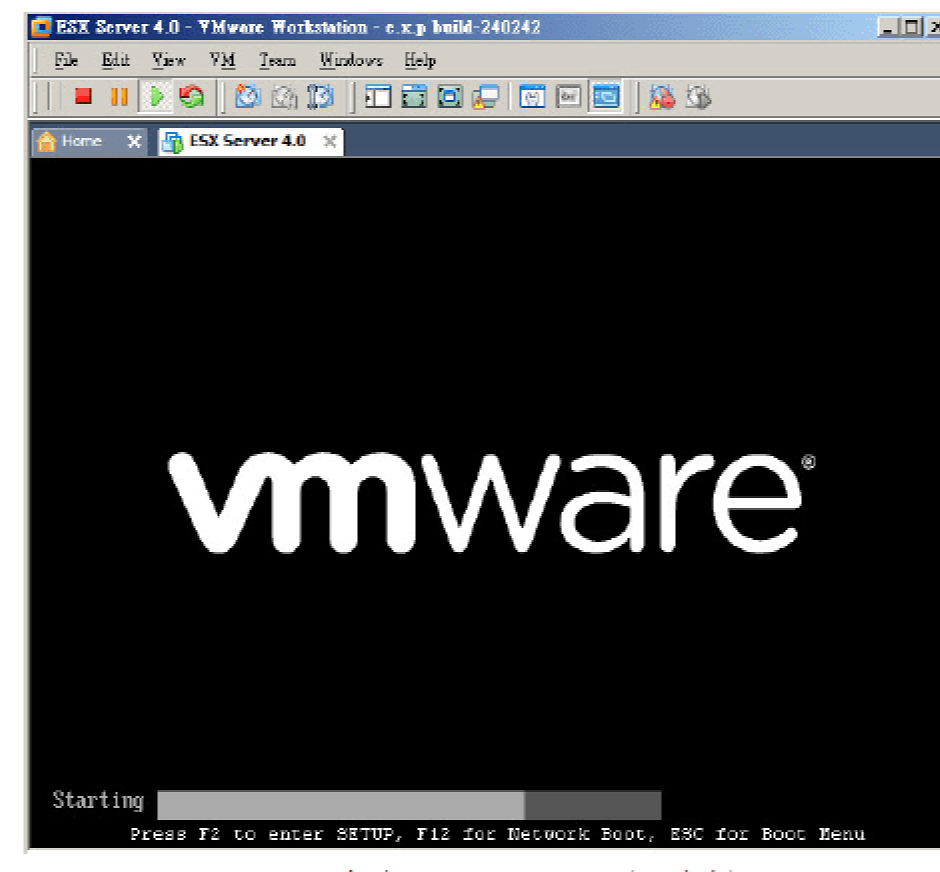
4. 开始安装 VM 中的 ESX/ESXi

安装 VM 中的 ESX/ESXi 和物理机一模一样，主要是操作 VMware Workstation 的引导。下面就是步骤。

1. 在 VMware Workstation 的主画面中，选择 Power on this virtual machine 选项。
2. 此时会弹出引导画面。

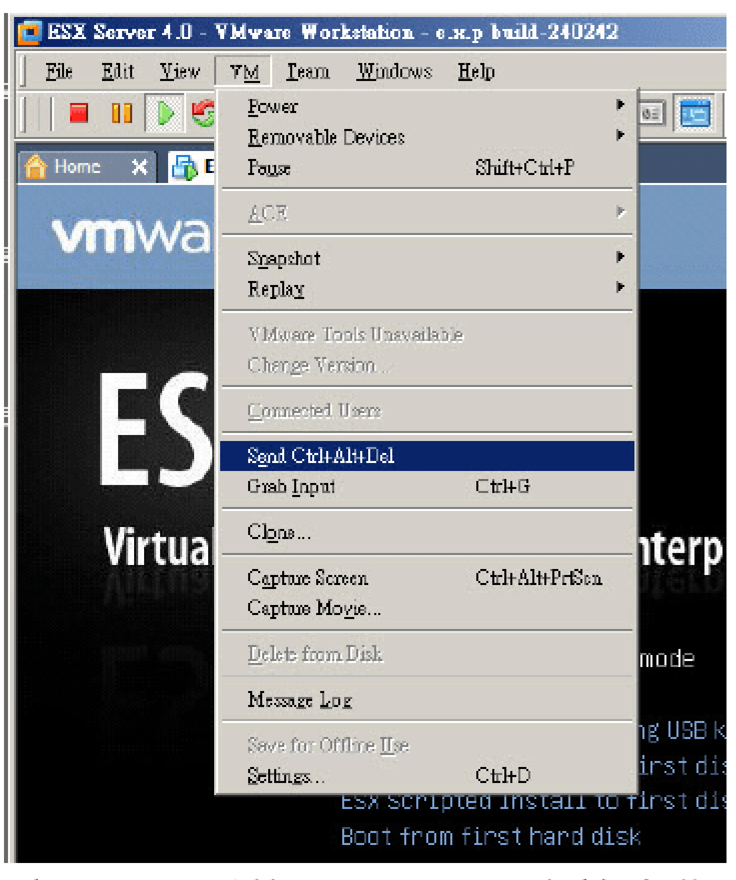


▲ 选择引导



▲ 弹出 VMware 的引导画面

3. 当弹出 ESX 的安装画面时，即可常规安装。
4. 由于这个窗口是一个 VM，因此如果要离开这个窗口，必须按 Ctrl+Alt 组合键才行，在这个 VM 中也可以安装 ESXi。



▲ 这是一个 VM，要按 Ctrl+Alt 组合键才能回到桌面上

结 语

事实上安装 ESX/ESXi 是整个 vSphere 中最简单的部分，因为就像安装任何一个 Linux 的操作系统一样。ESX/ESXi 的安装虽然简单，但却是构成 vSphere 整体架构的完整基础。安装完毕之后，接下来就是正规进入 ESX/ESXi 的使用了。

读书笔记

第 9 章

熟悉 ESX/ESXi 的基本使用

关键词：

- 如何从本地配置 ESXi 服务器
- 制作 USB 可引导 ESXi HypervCentersor
- ESX 的主控台简介
- 取得及安装 vSphere Client
- 创建第一台 ESX/ESXi 的虚拟机
- ESX/ESXi 中的快照功能（Snapshot）
- 和 ESX 主机交换数据

在安装好 ESX/ESXi 之后，从本地操作还无法使用虚拟机。虽然 ESX 可以用本地来管理虚拟机，但 VMware 建议是使用客户端的 vSphere Client。由于 vSphere Client 不但是管理单独 ESX/ESXi 的界面，在往后的 vCenter 架构中，vSphere Client 也是最佳的管理程序，因此在这一章中，我们就来看看 vCenter 的安装使用，并且利用 vCenter 来创建你的第一台 ESX 虚拟机。

9.1 ESXi 的基本功能

在正规安装 vCenter 之前，我们必须配置 ESX/ESXi 才能顺利让整个虚拟机架构方便使用。由于 ESX 本身就是完整的 Red Hat Linux 改良版，因此在 Linux 下的大部分文本命令（CLI）都可以使用。ESXi 是规约版的 ESX，所提供的命令并不完整，而其界面也相当简单，但我们还是可以通过其本身的界面来配置基本的参数。我们就先来看看较简单的 ESXi。

```
VMware ESX 4.0 (Kandinsky)
Kernel 2.6.18-128.ESX on an x86_64

ESXonWK7 login: root
Password:
Last login: Tue Apr 20 15:14:38 on tty1
[root@ESXonWK7 ~]# uname
Linux
[root@ESXonWK7 ~]# uname -a
Linux ESXonWK7 2.6.18-128.ESX #1 Thu Oct 15 16:11:16 PDT 2009 x86_64 x86_64 x86_
64 GNU/Linux
[root@ESXonWK7 ~]# _
```

▲ ESX 本身是一个标准的 Linux，命令都是标准的 C shell 或 bash Shell

```
        ESXi 4.0 http://www.vmware.com
        Copyright (c) 2007-2009 VMware, Inc.

Password:
You have activated Tech Support Mode.
The time and date of this activation have been sent to the system logs.

Tech Support Mode is not supported unless used in consultation
with VMware Tech Support.

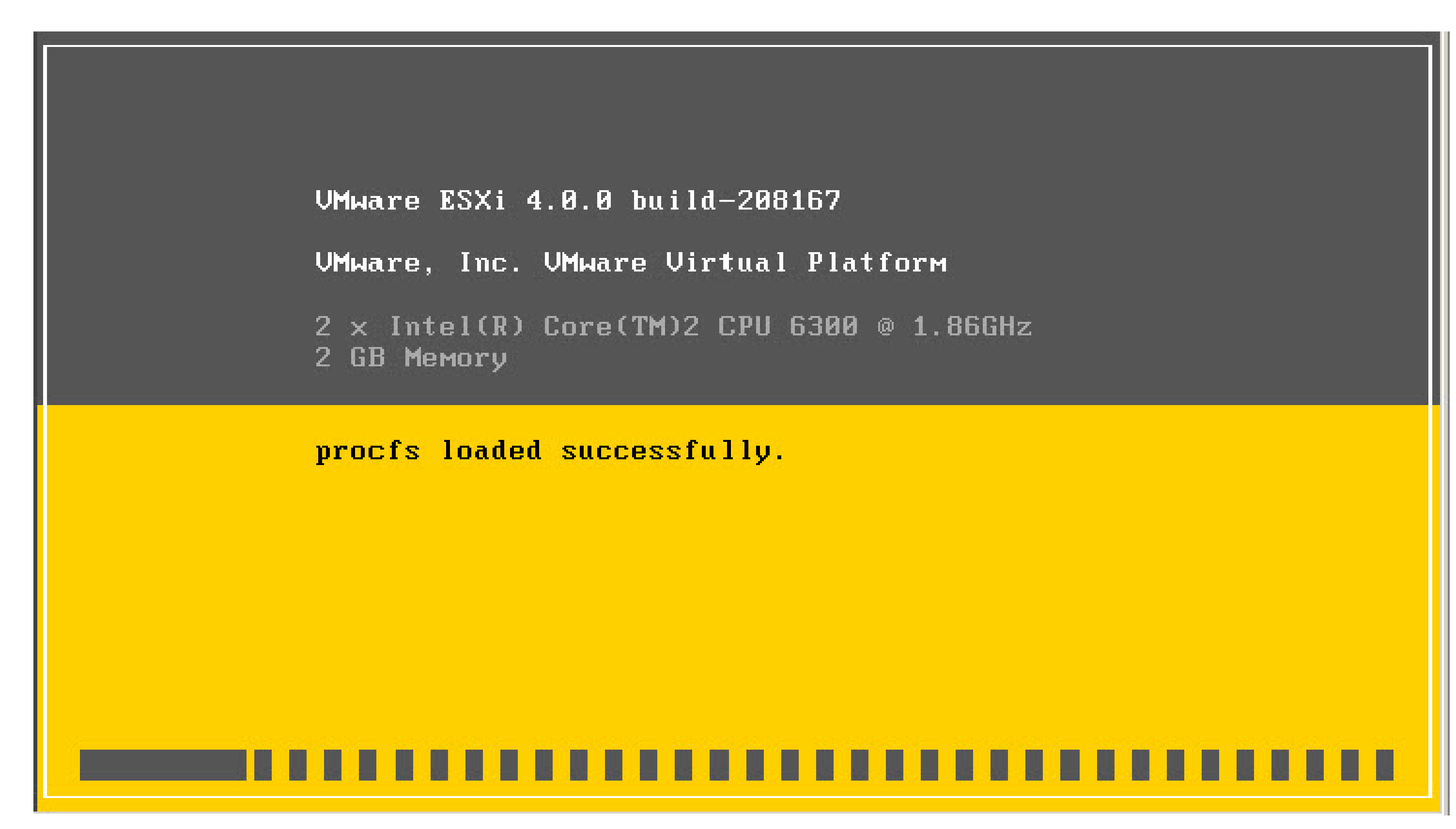
VMware offers supported, powerful system administration tools.  Please
see www.vmware.com/go/sysadmintools for details.

Tech Support Mode may be disabled by an administrative user.
Disabling requires a reboot of the system.  Please consult the ESXi
Configuration Guide for additional important information.
```

▲ 较为规约，但还是可以使用 SSH 连入（必须手动打开）

9.1.1 配置 ESXi 服务器

在安装并激活 ESXi 之后，首先我们会进入灰黄色的画面，这就是标准的 ESXi 的主控台。在主控台中能做的事不多，包括配置 IP 以及账号口令等。我们就先来看看这些最基本的任务。



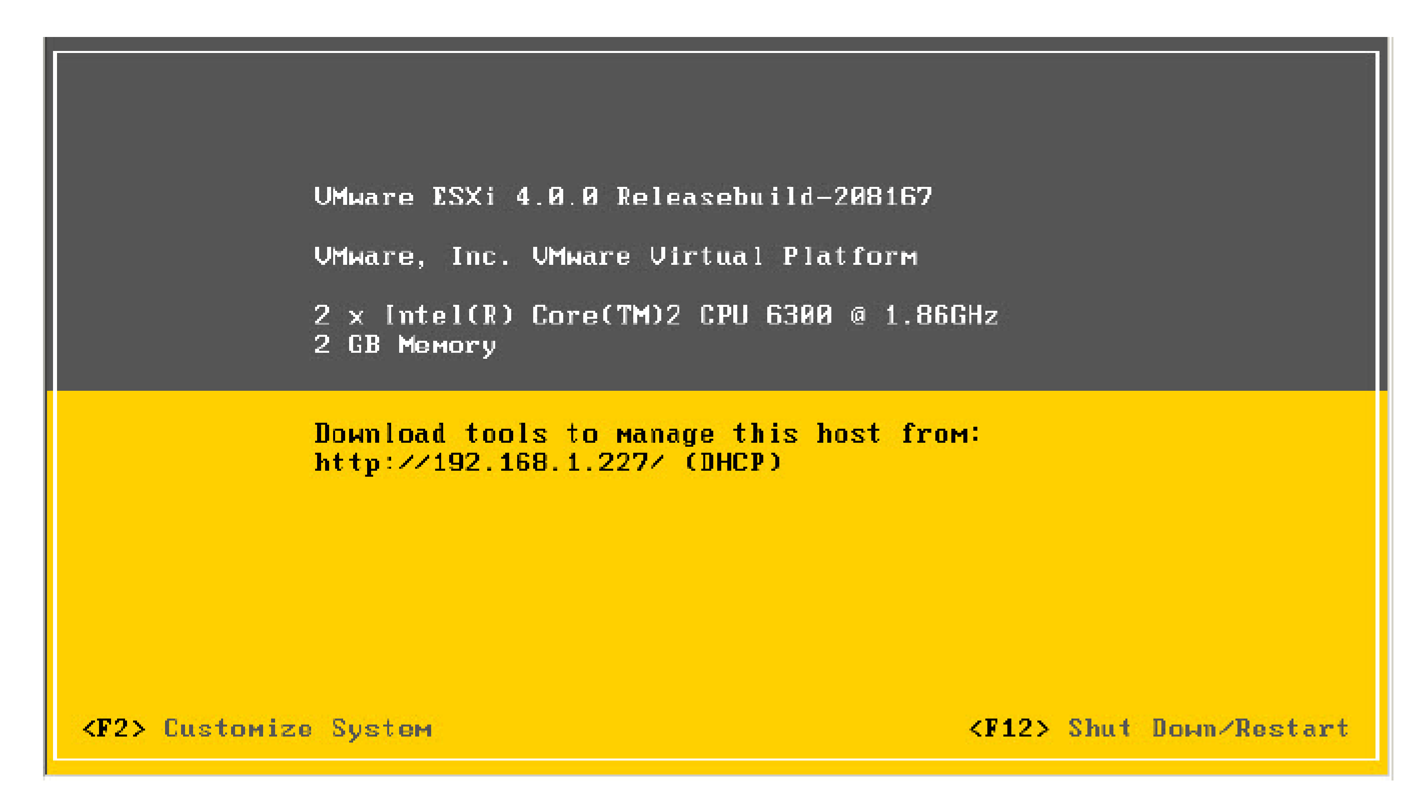
▲ ESXi 的标准装载画面

1. 配置 ESXi 的 IP 及账号口令

当我们进入 ESXi 之后，在灰黄色画面中会弹出一些基本的消息，包括了当前的 ESXi 版本编号、使用的 CPU 和内存大小以及 IP 参数等。这些参数都是可以通过主控台来修改的。

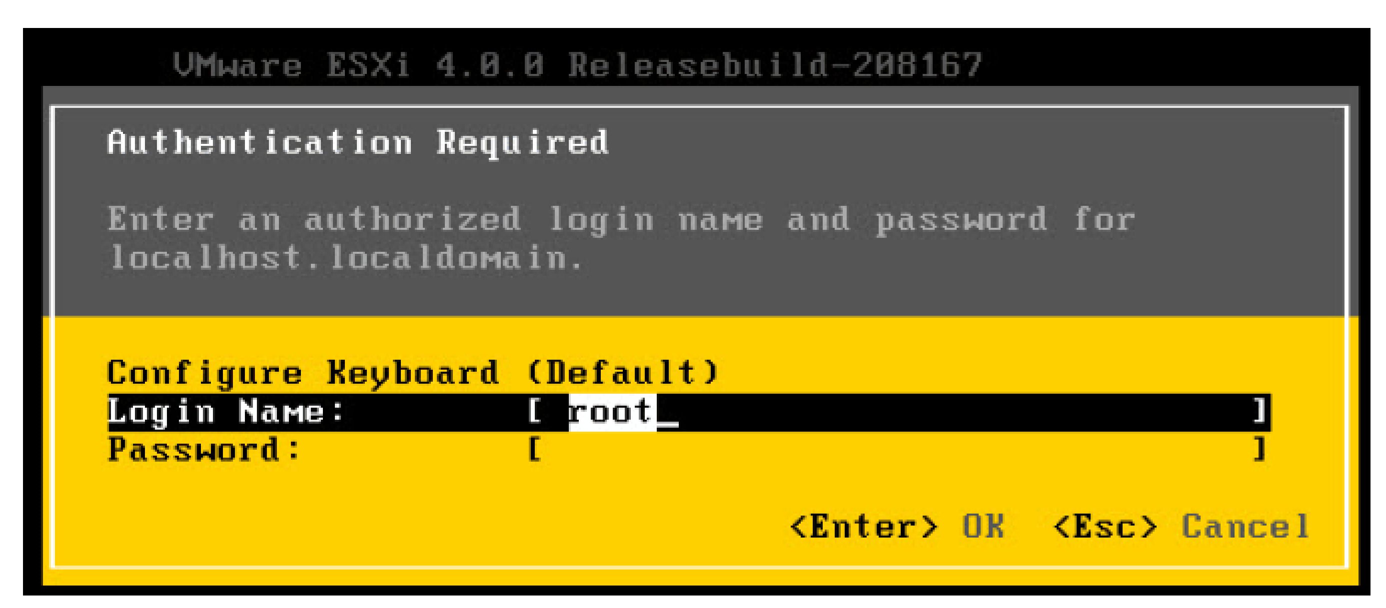
▶ 修改 ESXi 的 IP 参数及账号口令

1．在灰黄色的画面按 F2 键。



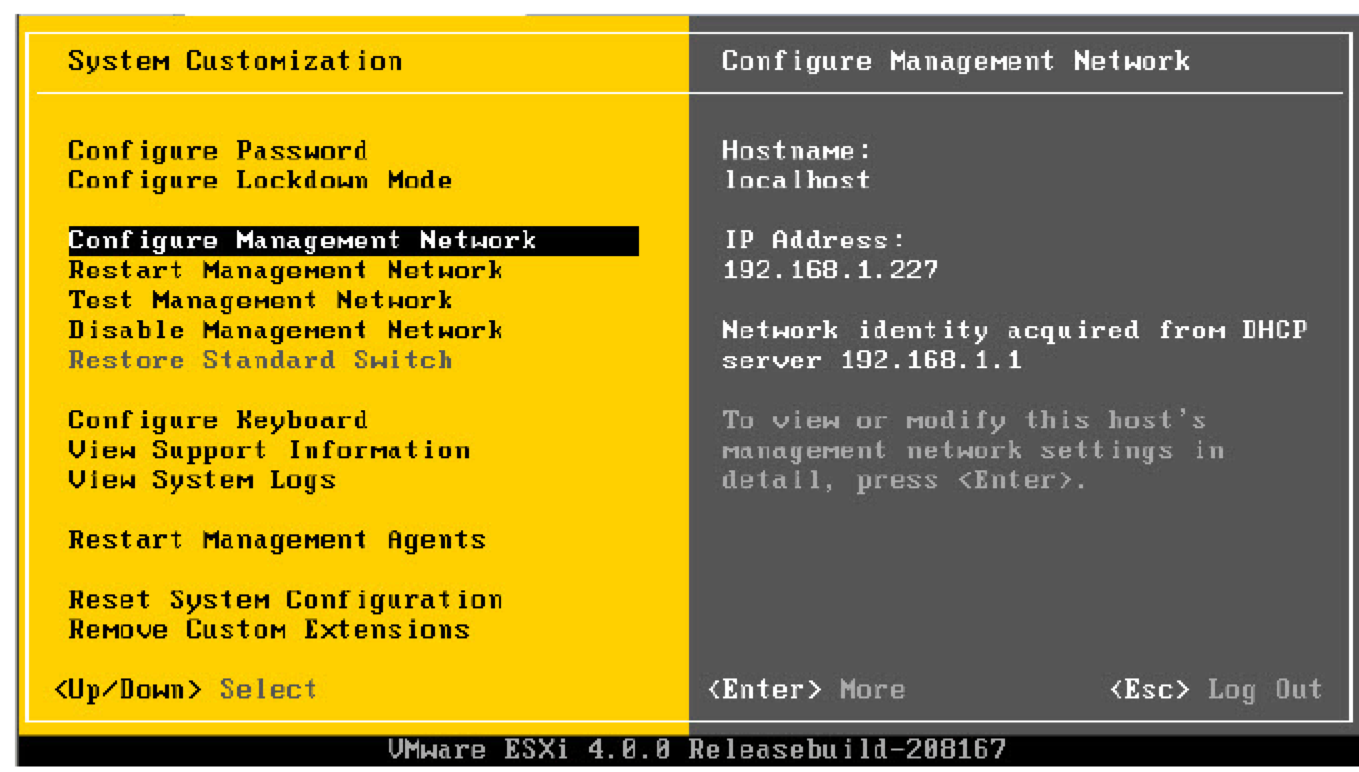
▲ 在此按 F2 键

2．在第一次配置的情况下，默认的账号是 root，口令为空。直接按 Enter 键即可。



▲ 账号口令的键入画面

3．此时会弹出 ESXi 的标准配置画面，较常使用的就是第一项的配置账号口令以及第三项的网络参数配置。



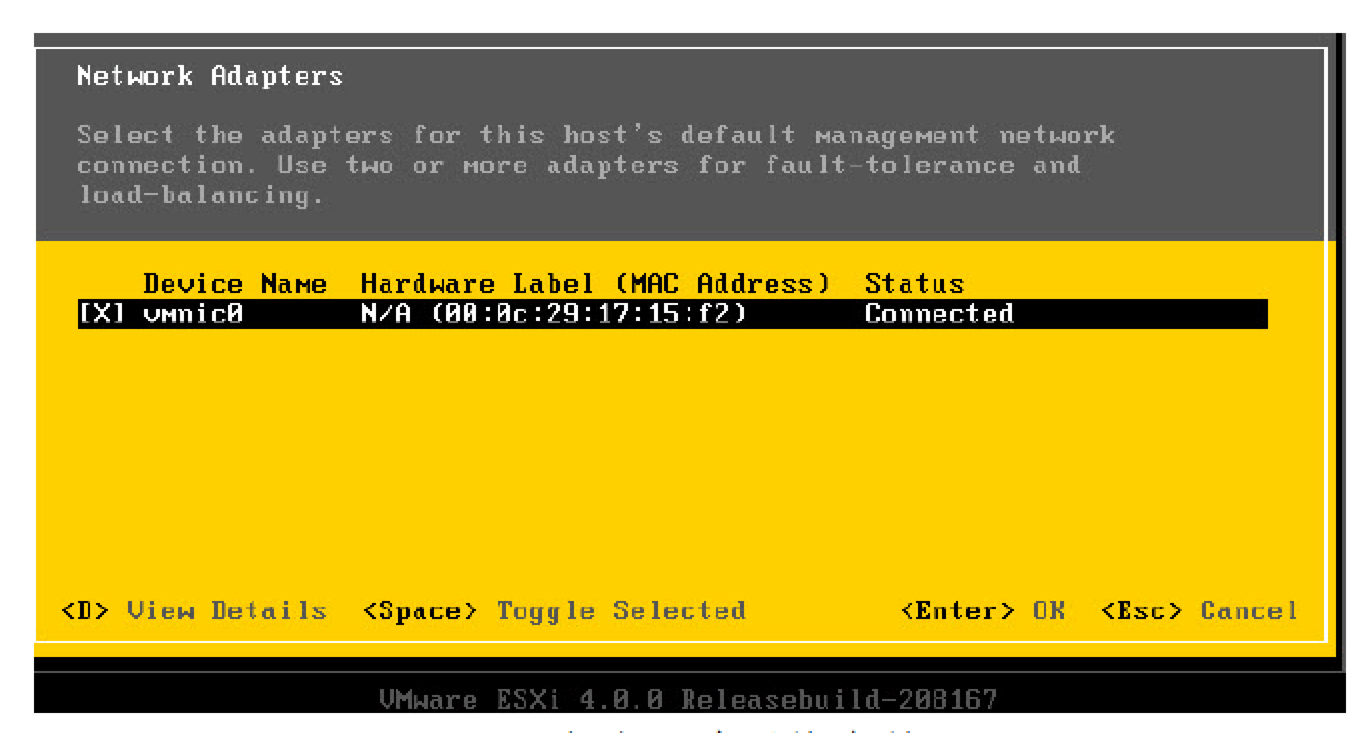
▲ 最常见的就是网络和账号配置参数

4．若要更改 root 账号的口令，进入第一项键入旧口令再键入两次新口令即可。



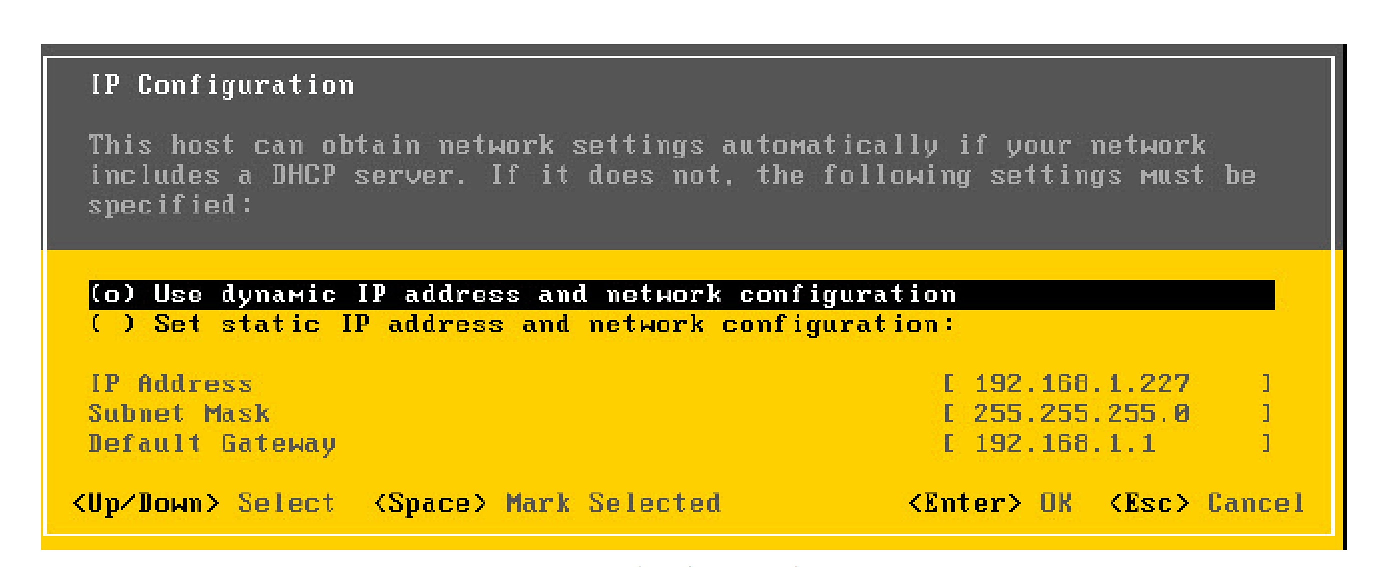
▲ 更改口令在此配置

5．如果要更改 IP 参数则是进入第三项。进入之后可看到 Network Adapters 选择给 ESXi 使用的网卡 vmnic0。



▲ 在此更改网络参数

6．回到前一个画面再进入 IP Configuration，则可以修改 IP 的使用方式，如 DHCP 或是手动配置 IP，建议使用静态的配置以方便 vSphere Client 使用。



▲ 修改 IP 参数

7．当配置完网络参数时，选择 Restart Management Network 会重新配置 IP 参数，从内网中就可以用 vSphere Client 连入了。



▲ 别忘了重新激活网络才会生效

2. 激活 ESXi 中的 SSH 连入

在大部分的企业级环境中，系统管理员是很少有机会常常进机房在服务器前面操作的。在

Windows Server 上我们会用远程遥控，而在 Linux 的主机则是用 SSH 来连入管理。ESXi 是一个以 Linux 为内核的 HypervCentersor 操作系统，因此也可以使用 SSH 来连入，但 ESXi 默认是不打开 SSH 的，因此我们必须手动打开。

▶ 手动打开 ESXi 的 SSH

1. 引导进入 ESXi 的主画面，按 Alt+F1 组合键。

```
VMware ESXi 4.0.0 Releasebuild-208167

VMware, Inc. VMware Virtual Platform

2 x Intel(R) Core(TM)2 CPU 6300 @ 1.86GHz
2 GB Memory

Download tools to manage this host from:
http://192.168.1.227/ (DHCP)

> Customize System                    <F12> Shut Down/Restart
```

▲ 在此按 Alt+F1 组合键

2. 此时会弹出如下图所示的画面，键入 unsupported。在键入的并发计算机是不会响应的，因此键入错误就必须重新引导了，键入完按 Enter 键。

```
       ESXi 4.0 http://www.vmware.com
       Copyright (c) 2007-2009 VMware, Inc.

Password:
You have activated Tech Support Mode.
The time and date of this activation have been sent to the system logs.

Tech Support Mode is not supported unless used in consultation
with VMware Tech Support.

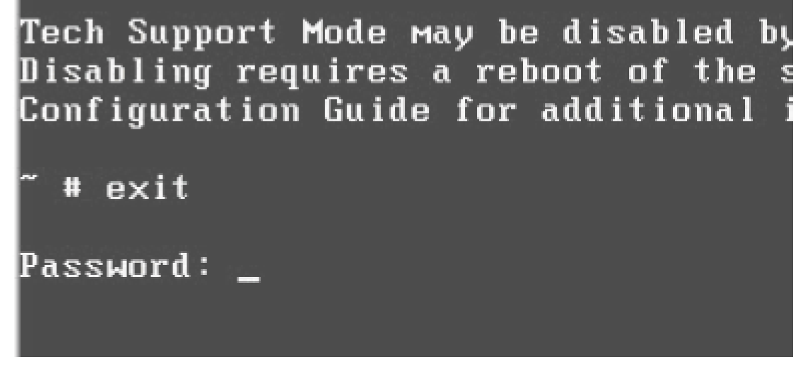
VMware offers supported, powerful system administration tools.  Please
see www.vmware.com/go/sysadmintools for details.

Tech Support Mode may be disabled by an administrative user.
Disabling requires a reboot of the system.  Please consult the ESXi
Configuration Guide for additional important information.

~ # exit
_
```

▲ 键入 unsupported

3. 此时会要求你键入 root 账号的口令，键入完按 Enter 键。



▲ 键入 root 账号的口令

4. 这时会进入 ESXi 的命令提示行。我们要做的就是更改 Linux 的引导配置文件，将 SSH 这个服务打开。

```
Password:
login: pam_authenticate call failed: Authentication failure (7)

Login incorrect
Password:
You have activated Tech Support Mode.
The time and date of this activation have been sent to the system logs.

Tech Support Mode is not supported unless used in consultation
with VMware Tech Support.

VMware offers supported, powerful system administration tools.  Please
see www.vmware.com/go/sysadmintools for details.

Tech Support Mode may be disabled by an administrative user.
Disabling requires a reboot of the system.  Please consult the ESXi
Configuration Guide for additional important information.

~ # _
```

▲ 进入 ESXi 的主控台

5. 键入 vi /etc/inetd.conf，就会进入编辑画面。

```
Tech Support Mode may be
Disabling requires a rebo
Configuration Guide for a

~ # vi /etc/inetd.conf_
```

▲ 编辑这个文件

6. 找到#ssh，并且将游标移到#ssh 的最第一个 s 上

7. 此时按下键盘上的 A 键（不分大小写），会进入编辑模式，按 Backspace 键将 #删除。

```
#echo     stream  tcp   nowait  root    internal
#echo     dgram   udp   wait    root    internal
#discard  stream  tcp   nowait  root    internal
#discard  dgram   udp   wait    root    internal
#daytime  stream  tcp   nowait  root    internal
#daytime  dgram   udp   wait    root    internal
#chargen  stream  tcp   nowait  root    internal
#chargen  dgram   udp   wait    root    internal
#time     stream  tcp   nowait  root    internal
#time     dgram   udp   wait    root    internal

# Remote shell access
#
#ssh    stream  tcp     nowait  root    /sbin/dropbearmulti
#ssh    stream  tcp6    nowait  root    /sbin/dropbearmulti
#telnet stream  tcp     nowait  root    /bin/busybox    tel
#telnet stream  tcp6    nowait  root    /bin/busybox    tel

# VMware authentication daemon
- /etc/inetd.conf 31/40 77%
```

▲ 主要是要将#拿掉

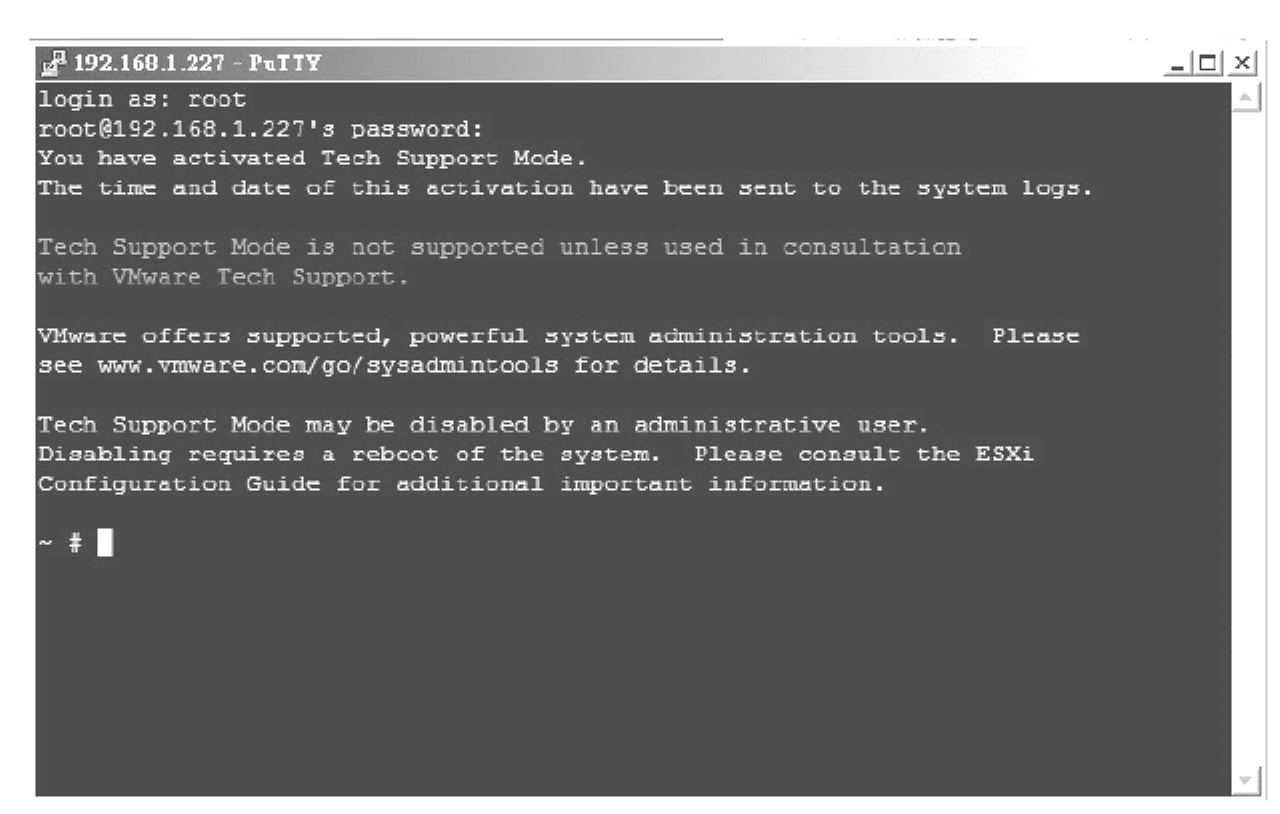
8. 按:键回到 vi 的命令模式，并且键入 wq 将变更存盘并离开，并且重新激活 ESXi 服务器。

```
# Remote shell access
#
ssh     stream  tcp     nowait  root    /sbin/dropbea
#ssh    stream  tcp6    nowait  root    /sbin/dropbea
#telnet stream  tcp     nowait  root    /bin/busybox
#telnet stream  tcp6    nowait  root    /bin/busybox

# VMware authentication daemon
~ # reboot
~ # Running sfcbd-watchdog stop
Running wsman stop
Stopping openwsmand
Running slpd stop
Stopping slpd
Running vobd stop
watchdog-vobd: Terminating watchdog with PID 4950
Vobd stopped.
Running hostd stop
watchdog-hostd: Terminating watchdog with PID 4928
Running ntpd stop
Stopping ntpd
```

▲ 回到命令列并且存档离开

9. 进入一个有 SSH 客户端的计算机，键入 ssh 192.168.1.227 （即 ESXi 的固化 IP）。如果能成功连入就意味着配置成功了。



▲ 可以顺利连入 ESXi 了

9.1.2 制作移动的 USB 可引导 ESXi

ESXi 的轻薄短小让许多中小型企业也对虚拟机这一个产品跃跃欲试。虽然 ESXi 十分规约，但麻雀虽小五脏俱全，所有 ESX 该有的虚拟机功能 ESXi 都有，而 ESXi 更有一个 ESX 没有的功能，就是你可以将 ESXi 制作成可引导的 USB HypervCentersor，只要服务器是在 VMware ESX/ESXi 的 HCL 支持范围，就可以随身携带 HypervCentersor，到哪都可以引导使用虚拟机的 HypervCentersor 了！

1. 准备任务

首先你必须有一个 1GB 以上的 USB 硬盘，并且可以引导。较新的主板（2004 年以后）都支持 USB 引导，你必须在 BIOS 中配置由 USB 引导。在准备好 USB 之后，还需要一个能解开 ISO 的解压缩程序（如 Winrar 或 zip、7zip 等）和能将 Linux dd 格式文件解开的程序，我们选择 WinImage。下面先给出需要的硬件和软件。

▶ **将 ESXi 解压 USB 的准备任务**

- 可以引导，1GB 以上的 USB 硬盘。
- 解 ISO 压缩文件案的工具软件 Winzip 或 Winrar、7Zip。
- WinImage。
- ESXi 原始 ISO 档。

2. 将 ISO 档转换至 USB 中

在准备好所有的软硬件之后，接下来就是将 ISO 中有关 USB 引导的部分提取出来，并且利用软件将引导程序克隆到 USB 中了。

▶ **将 ESXi 的 ISO 启动文件转入 USB 中）**

1. 将 USB 先插入计算机中，确定 USB 的磁盘代号。

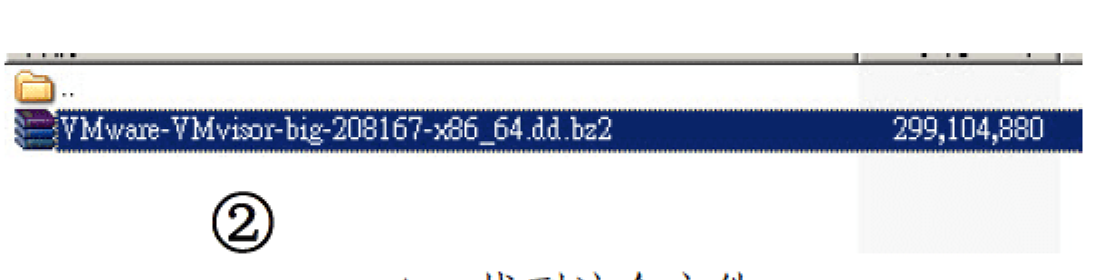


▲ 确定 USB 硬盘的代号

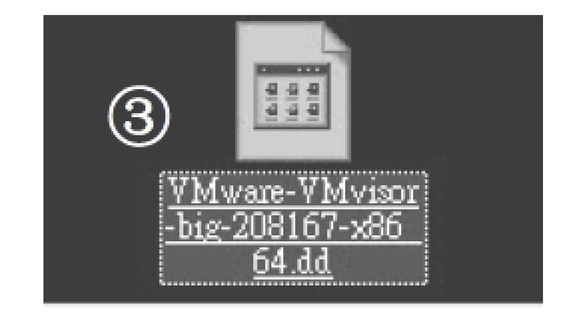
2. 首先我们用 Winrar 打开 ESXi 的原版 ISO 文件，解开之后会在 image.tgz 下，\INSTALL\usr\lib\vmware\

的目录下找到 VMware-VMvCentersor-big-208167-x86_64.dd.bz2 这个文件。

3．将 VMware-VMvCentersor-big-208167-x86_64.dd.bz2 中的 VMware-VMvCentersor-big-208167-x86_64.dd 文件解出来。



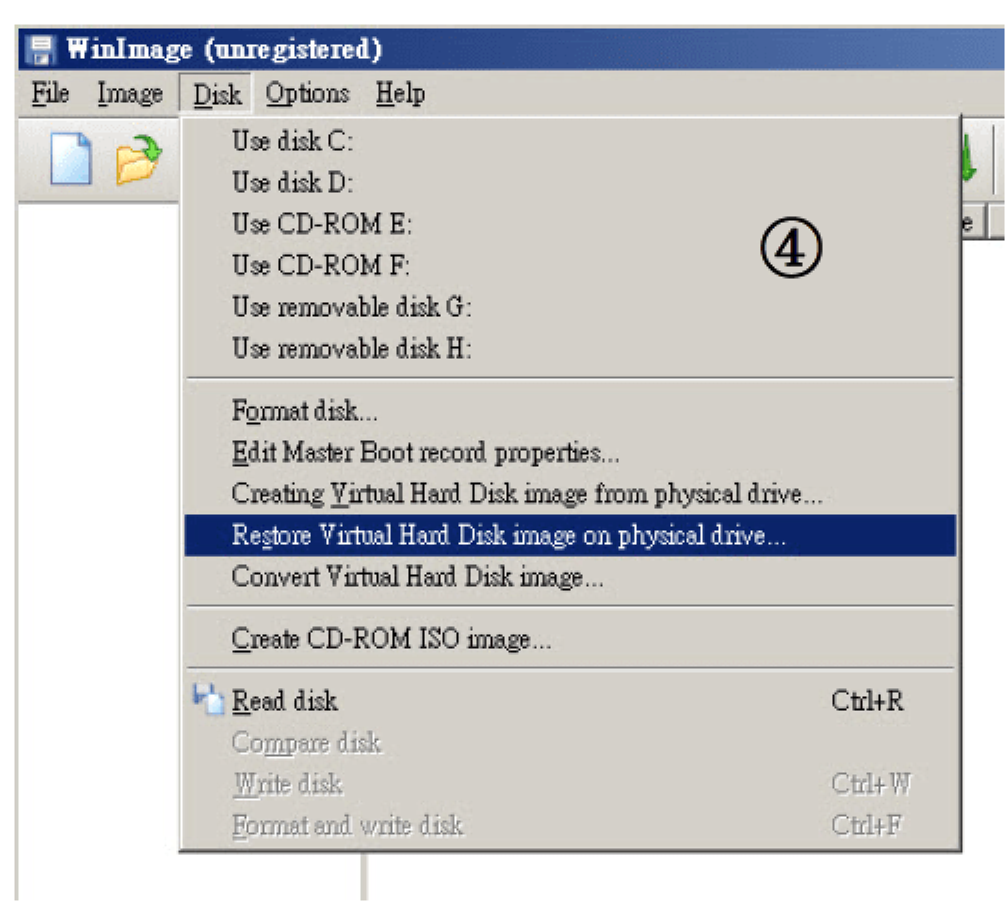
▲ 找到这个文件



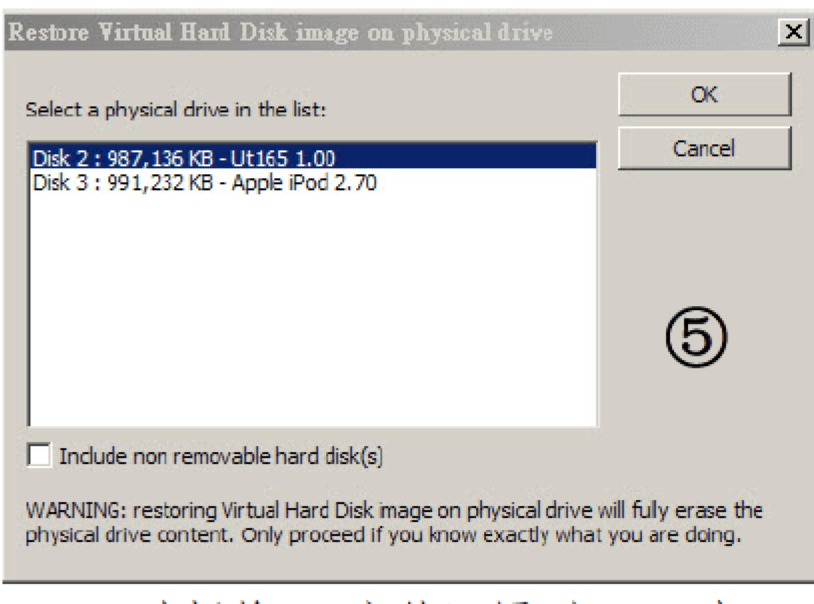
▲ 这是 Linux 格式的引导档

4．此时进入 WinImage 中，选择 Disk/Restore Virtual Hard Disk image on physical drive...选项。

5．选择 USB 磁盘，并且选择刚才解开的 dd 文件。



▲ 选择还原图像的部分



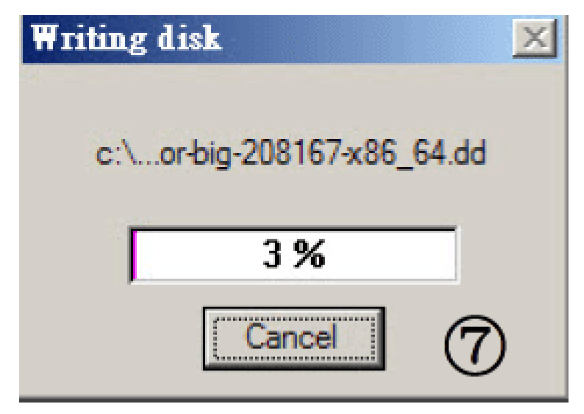
▲ 选择将 dd 文件还原到 USB 中

6．当弹出如下左图所示的画面时，单击“是（Y）”按钮进行还原。

7．当弹出如下右图所示的画面时，意味着 ESXi 已经开始还原，只要主机能匹配 VMware 的 HCL，并且支持 USB 引导，这台计算机就可以用来作 ESXi 的服务器了。



▲ 进行还原



▲ 等待还原完毕，这个 USB 就可以引导，成为可携式的 HypervCentersor 了

9.2 配置 ESX 服务器

ESX 相对于 ESXi 较为复杂，但系统的复杂带来更多的好处。由于 ESX 的总控部分本身就贴

近一个完整的 Linux，因此在 ESX 上的操作中，除了和虚拟机相关的命令，大部分的命令就是 Linux 的 C shell 或是 bash 的命令。如果读者们对 Linux 不熟悉的话，可以参阅网络或本出版社和 Linux 有关的书籍或文章。

9.2.1 ESX 的主控台简介

ESX 的本地操作界面称为主控台（ServCenterce Console），这是 ESXi 所缺少的。ESX 的主控台功能相当完整，可以在这里进行所有和虚拟机有关的操作。ESX 中提供了相当复杂且繁多的命令集针对虚拟机以及相关资源进行管理和总控，因此也可以编写 Script 文件做批量管理。

由于在 ESX 下，网络、CPU 等设备都被 HypervCentersor 接管，因此在 ESX 的主控台之下，原来 Linux 管理网络或系统资源的命令全部被 ESX 的命令取代，因此和网络以及资源、VMFS 格式硬盘有关的命令，将无法再使用 Linux 下的命令管理，全部必须用 ESX 提供的虚拟机管理了。我们首先就来看最常用的 ESX 命令。

1. 变更 ESX 的 IP 参数

ESX 将网卡收回给 HypervCentersor 管理，因此其主控台下的网卡也算是一个虚拟的网卡，并且联机到虚拟的路由器上。这个观念和 Hyper-V 的虚拟路由器十分贴近，因此在主控台中总控的网卡，并不是这台机器的真实网卡信息。在 ESX 中，默认在主控台中使用的网络界面称为 vswif0，因此我们要更改网络 IP 参数时，就是更改这个 vswif0 的参数。而更改网络参数相当于更动 HypervCentersor 所掌管的系统资源，因此必须使用 ESX 的命令而非 Linux 的命令。

▶ 更改 ESX 的 ServCenterce Console IP

1. 进入 ESX 的主控台，在下图所示画面下按 Alt+F1 组合键可以激活主控台，在键入账号口令后就可以进入命令行。

```
VMware ESX 4.0 (Kandinsky)
Kernel 2.6.18-128.ESX on an x86_64

ESXonWK7 login:    Starting VMs (background)              [  OK  ]


VMware ESX 4.0 (Kandinsky)
Kernel 2.6.18-128.ESX on an x86_64

ESXonWK7 login: _
```

▲ 进入 ESX 的主控台

2. 由于我们在安装 ESX 时已经键入一个 IP 了，因此在主控台必须先将这个 IP 代表的网络界面删除。在命令行下键入 esxcfg-vswif –d vswif0。这个动作是将原来的网络界面删除。如果没有提示错误消息，意味着删除成功。

```
[root@ESXonWK7 ~]#
[root@ESXonWK7 ~]#
[root@ESXonWK7 ~]# esxcfg-vswif -d vswif0
Nothing to flush.
[root@ESXonWK7 ~]# _
```

▲ 先将安装时配置的网卡删除

3．接下来我们要将新的网络界面加入 ESX 中键入 esxcfg-vswif –a vswif0 –p Service\ Console –i 192.168.1.88 –n 255.255.255.0 –b 192.168.1.255。其中 192.168.1.88 是新的IP，255.255.255.0 是 subnet mask，192.168.1.255 是广播 IP。

```
[root@ESXonWK7 ~]#
[root@ESXonWK7 ~]#
[root@ESXonWK7 ~]# esxcfg-vswif -d vswif0
Nothing to flush.
[root@ESXonWK7 ~]# esxcfg-vswif -a vswif0 -p Service\ Console -i 192.168.1.88 -n
 255.255.255.0 -b 192.168.1.255
[2010-04-21 11:08:37 'Vnic' warning] Generated New MAC address, 00:50:56:44:fc:6
4 for vswif0
Nothing to flush.
[root@ESXonWK7 ~]# _
```

▲ 利用 ESX 的命令新加入一个网络界面并且配置 IP

4．当配置完毕之后，我们可以键入 Linux 的命令 ifconfig，就会弹出当前的网卡配置和其 IP。

```
          TX packets:78 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:4388443 (4.1 MiB)  TX bytes:5040 (4.9 KiB)
          Interrupt:145

vswif0    Link encap:Ethernet  HWaddr 00:50:56:44:FC:64
          inet addr:192.168.1.88  Bcast:192.168.1.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:8 errors:0 dropped:0 overruns:0 frame:0
          TX packets:6 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:480 (480.0 b)  TX bytes:252 (252.0 b)
```

▲ ifconfig 是传统的 Linux 命令，不是 ESX 命令，可以查看 IP

5. 接下来要更改 Gateway 和主机名，我们直接更改配置文件。键入 vi /etc/sysconfig/network 就会进入更改 Gateway 的文件。

```
[root@ESXonWK7 ~]#
[root@ESXonWK7 ~]# vi /etc/sysconfig/network_
```

▲ 又用到 vi 了，读者们一定要熟悉 vCenter 的操作

6．这个文件中相当自觉，你可以更改主机名或是 Gateway，先将光标移到要更改的行上，按键盘 A 键，就可以更改了。

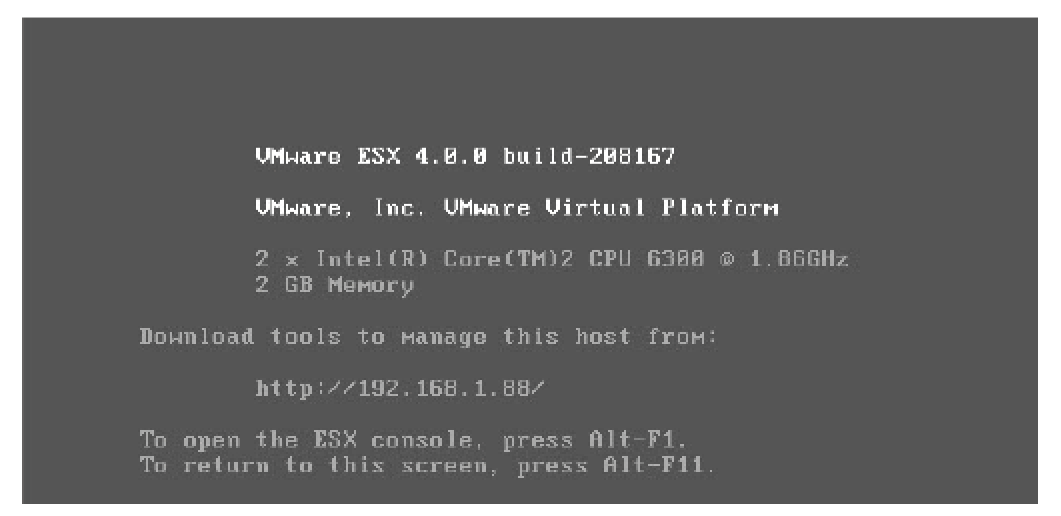
7．更改完按 Esc 键会离开编辑模式，然后按:键会进入命令模式，键入 wq 会将数据存档并离开 vCenter。

8．并发也修改/etc/hosts 文件，不然会在引导时仍然显示旧的 IP。

9．之后我们键入 reboot 重启，引导之后就可以看到 IP 已经变更了。

```
127.0.0.1        localhost
::1              localhost
192.168.1.88     ESXonWK7
~
~
~
~
```

▲ 也要修改/etc/host 档，以免引导时还会显示旧的 IP



▲ IP 已更改了

> **注意**
>
> **学习 vi 的使用**
>
> 如果读者们对 Linux 下的编辑程序 vi 并不熟悉，现在是学习 vi 的好时机。由于 ESX 或 ESXi 不管 Linux 的内核规约，是不可能将 vi 移出使用界面的。vCenter 是老牌的 Linux 下编辑软件，所有版本的 Linux 都有 vi 使用。本书中所有和 Linux 有关的编辑都是使用 vi，因此想将 ESX 完全理解，vi 的使用是必须的。

2. 开放 ESX 的 SSH 连入

ESX 既然是标准的 Linux，当然应该允许 SSH 的连入。然而 ESX 的默认值是不让 SSH 以 root 的身份连入的。如果你常常需要用 SSH 连入 ESX，必须将 root 的 SSH 权限打开。我们就来看看详细的步骤。

▶ 将 ESX 下 root 的 SSH 权限打开

1．在本地连入 ESX 的主控台，并且用 root 账号登录。

```
VMware ESX 4.0 (Kandinsky)
Kernel 2.6.18-128.ESX on an x86_64

ESXonWK7 login: root
Password:
Last login: Wed Apr 21 11:07:09 on tty1
[root@ESXonWK7 ~]# _
```

▲ 选择用 root 登录

2．编辑 SSH 的配置文件，键入 vi/etc/ssh/sshd_config。

3．此时在文中找到 PermitRootLogin no 这一行，改成 PermitRootLogin yes，并且存档。

```
#       $OpenBSD: sshd_config,v 1.73 2005/12/06 22:38:28 reyk Exp $

# This is the sshd server system-wide configuration file.  See
# sshd_config(5) for more information.

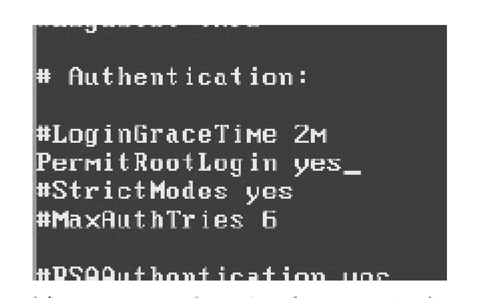
# This sshd was compiled with PATH=/usr/local/bin:/bin:/usr/bin

# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented.  Uncommented options change a
# default value.

#Port 22
#Protocol 2,1
Protocol 2
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::

# HostKey for protocol version 1
#HostKey /etc/ssh/ssh_host_key
# HostKeys for protocol version 2
#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_dsa_key
"/etc/ssh/sshd_config" 110L, 3299C
```

▲ 编辑这个文件



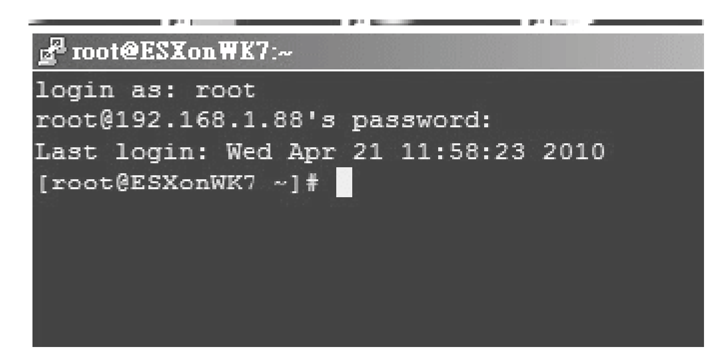
▲ 将 root 的登录配置为 yes

4．重新激活 SSH 服务，键入 service sshd restart。

5．此时你用 SSH 的程序就可以连入了。

```
[root@ESXonWK7 ~]#
[root@ESXonWK7 ~]#
[root@ESXonWK7 ~]# service sshd restart
Stopping sshd:                                             [  OK  ]
Starting sshd:                                             [  OK  ]
[root@ESXonWK7 ~]# _
```

▲ 别忘了重新激活 SSH 服务



▲ 可以连入了

9.3 安装及使用 vSphere Client

ESX/ESXi 的操作主要都是在 vSphere Client 之下，因此在安装完 ESX/ESXi 之后，要能顺利操作 ESX/ESXi，大部分的情况都是借由 vSphere Client 来进行。接下来我们就来看看 vSphere Client 的取得、安装以及基本的使用。

9.3.1 取得及安装 vSphere Client

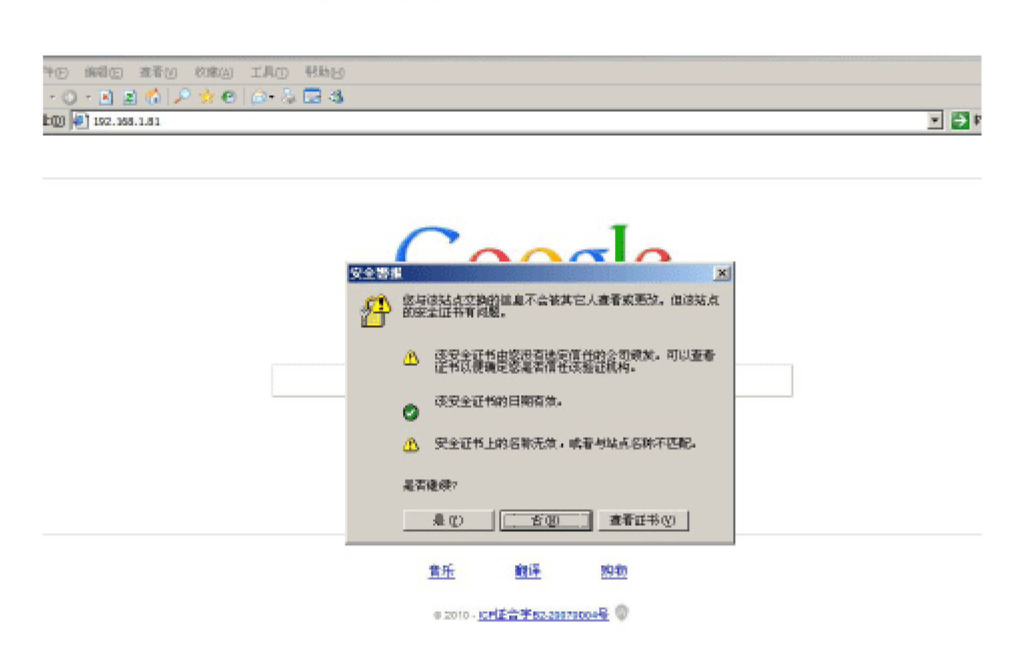
取得 vSphere Client 不需要到 VMware 的网站下载。当你安装好 ESX/ESXi 时，这个安装文件就已经在 ESX/ESXi 的服务器上了，我们只要将安装好的服务器取回即可。另外安装 vSphere Client 也十分简单，就是一般的 Windows 实用程序，只是在安装时需要注意一些配置的地方。

1. 取得 vSphere Client

在安装好 ESX/ESXi 之后，在主控台就会看到一个 IP 位置。这就是 ESX/ESXi 的管理 IP。要进行管理必须取得 vSphere Client，我们只要使用浏览器连入自身安装好的 ESX/ESXi 再进行下载即可。

▶ 下载 vSphere Client

1. 进入任何计算机的浏览器，并且键入http://192.168.1.211，其中该 IP 为你安装 ESX/ESXi 的 IP 位置。
2. 此时会弹出异常网站。如果是使用 IE，则直接点选进入网站，如果是使用 Firefox，则必须添加异常网站。



▲ 键入之后会弹出异常网址



▲ 使用 Firefox 会需要新建异常网站较麻烦

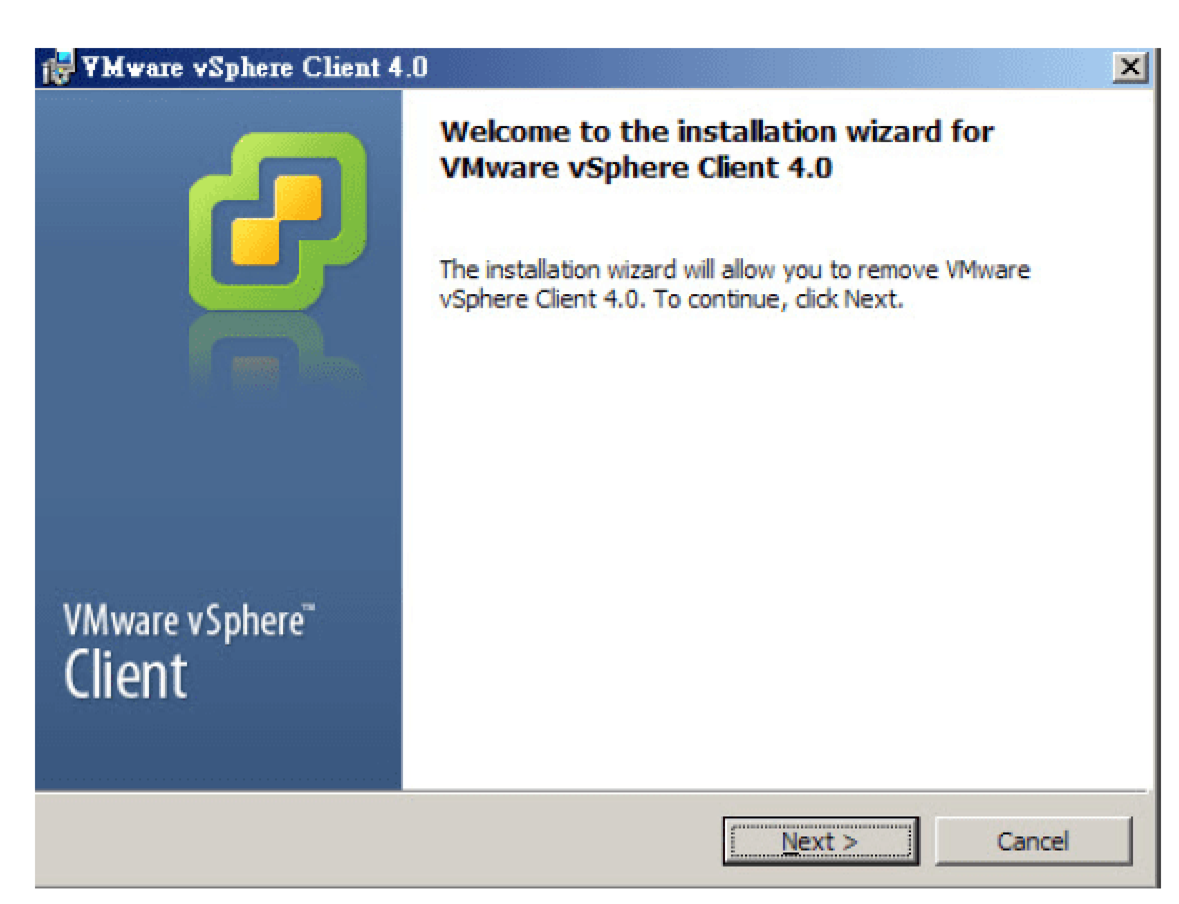
3. 进入之后，点选下载 vSphere Client 的链接，就开始下载了。



▲ 下载 vSphere Client

2. 安装 vSphere Client

vSphere Client 下载回来之后就可以直接安装了。但在安装 vSphere Client 之前，必须先安装微软的.NET Framework。如果你没有安装，系统在安装的过程也会替你安装，我们只要照着安装的步骤一步步安装即可，在此就不再详细介绍。



▲ 就像安装任何一个 Windows 实用程序一样

3. vSphere Client 连入 ESX/ESXi

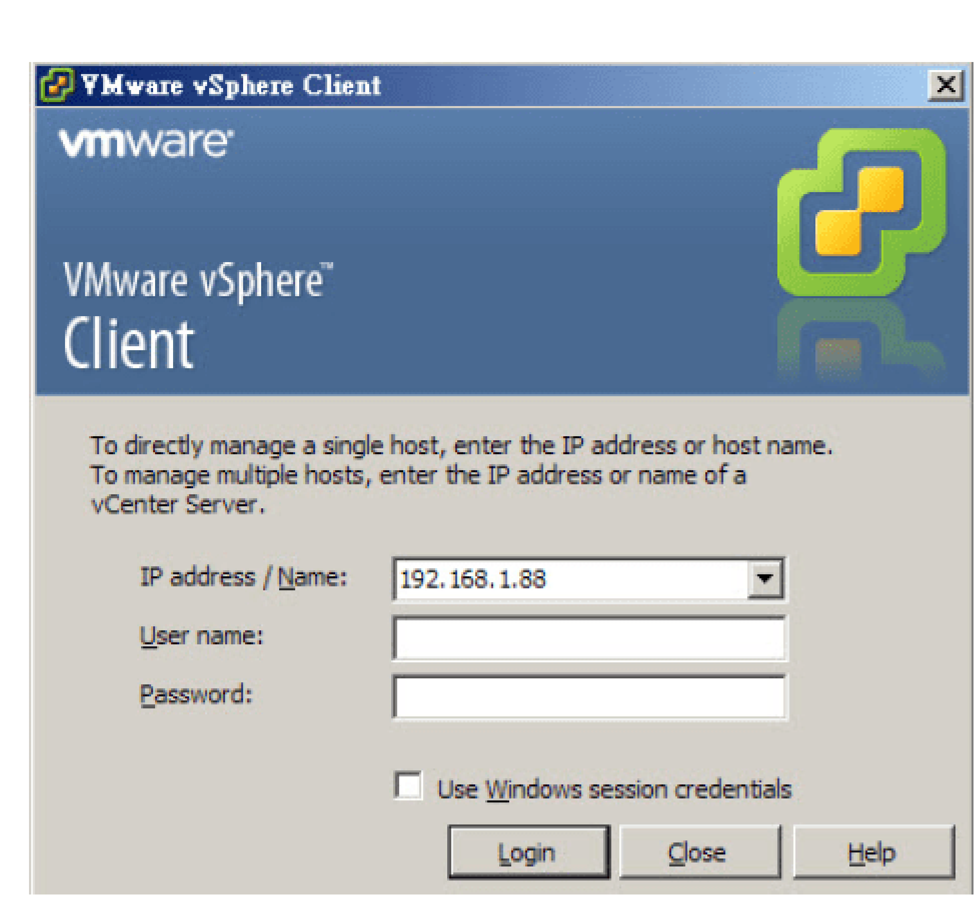
当安装好 vSphere Client 之后，就可以连入 ESX/ESXi 的服务器了。由于默认安装并没有使用 SSL 等安全的连接，会有报警消息，但不会影响使用，整个过程十分简单，我们就来看看用 vSphere Client 连入 ESX/ESXi 的方法。

▶ 从 vSphere Client 连入 ESX/ESXi

1. 直接单击桌面上安装好的 vSphere Client。
2. 此时会进入联机画面。键入 ESX/ESXi 的 IP、用户账号和口令，单击 Login 按钮即可开始联机。

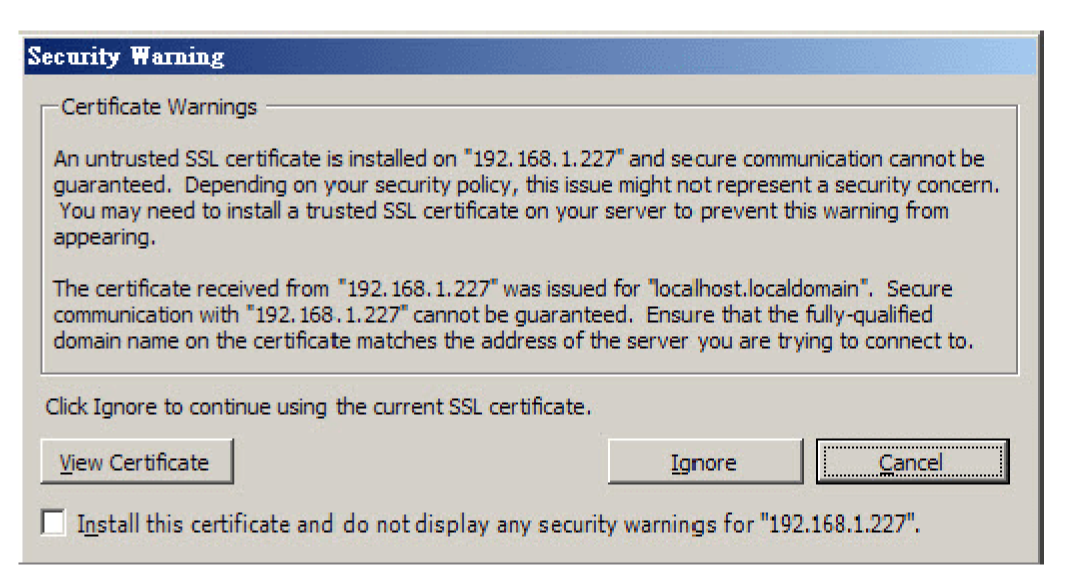


▲ 安装好会在桌面上创建一个图示



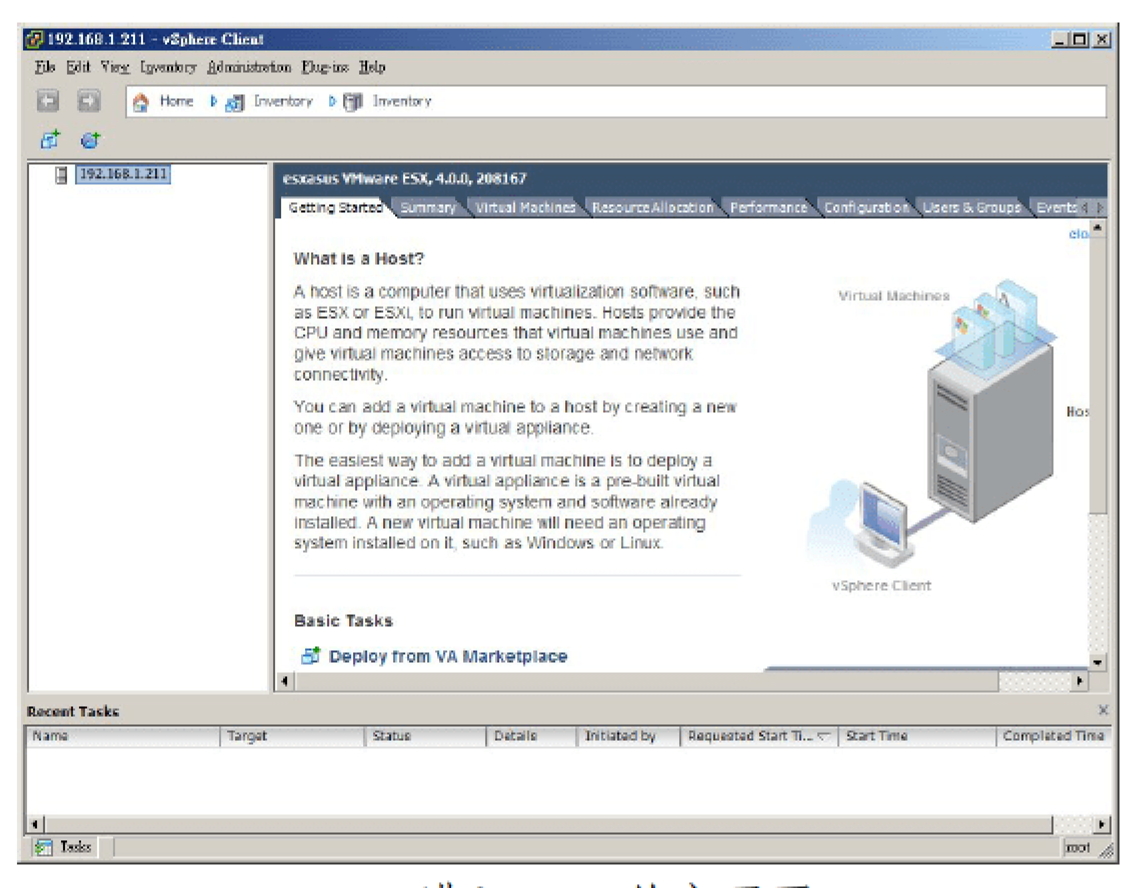
▲ 键入 IP 和账号口令登录

3. 由于没有使用 SSL，会先弹出一个报警窗口，单击 Ignore 按钮，也可以选中下方的选项，以后针对这个 ESX 就不会弹出报警窗口。



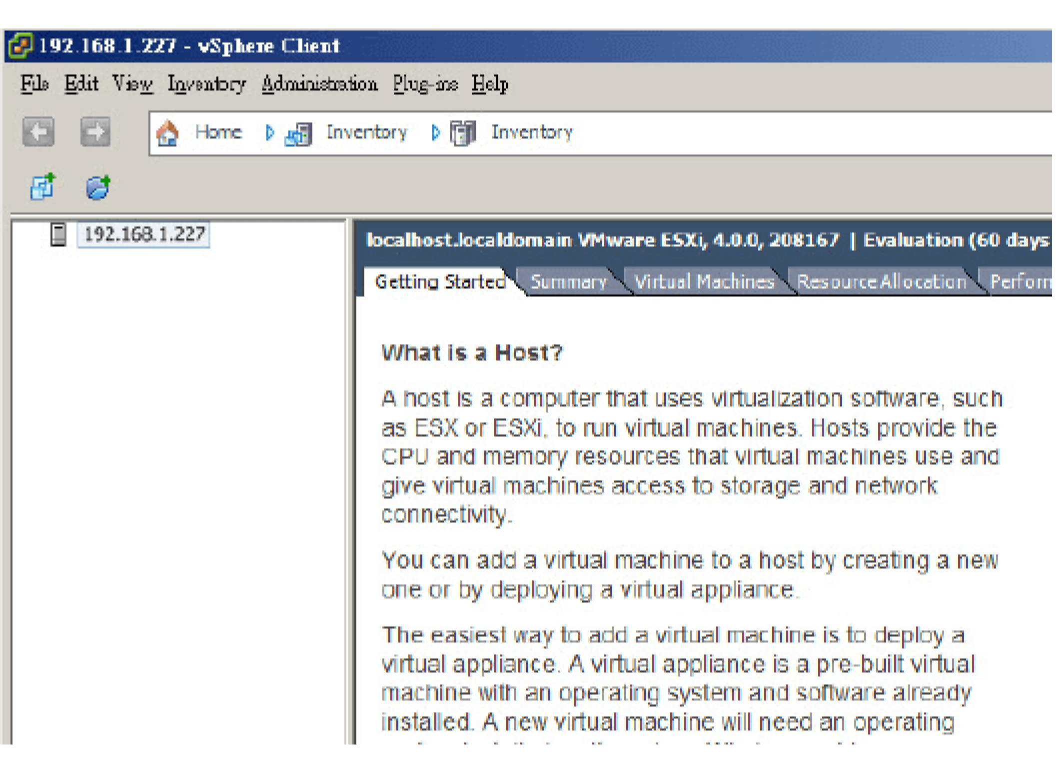
▲ 默认使用 SSL 联机，这里没有安装 SSL，因此会有一个报警窗口

4．当弹出的画面，意味着已经成功联机至 ESX 了。



▲ 进入 ESX 的主画面

5．下图所示则是连入 ESXi 的画面，使用 vSphere Client，进入 ESX/ESXi 两者的界面是完全的一样的。



▲ 连入 ESXi 也是一样的

9.3.2 创建第一台 ESX/ESXi 的虚拟机

vSphere Client 的功能十分强大，任何单物理机的虚拟机任务都可以从 vSphere Client 上落实。由于 ESX/ESXi 也涉及 Linux 权限的问题，因此在 vSphere Client 上，甚至可以进行用户权力的管理。我们会在下一章，针对 ESX/ESXi 有更详细的说明。在这一章中，我们就先从创建 ESX/ESXi 虚拟机的具体示例来进行。

▶ 在 ESX 中创建虚拟机的步骤

（1）创建一个文件夹来存放未来的虚拟机并上传 ISO 文件。

（2）创建一个虚拟机，包括各种参数。

（3）将光盘机给定到上传的 ISO 文件。

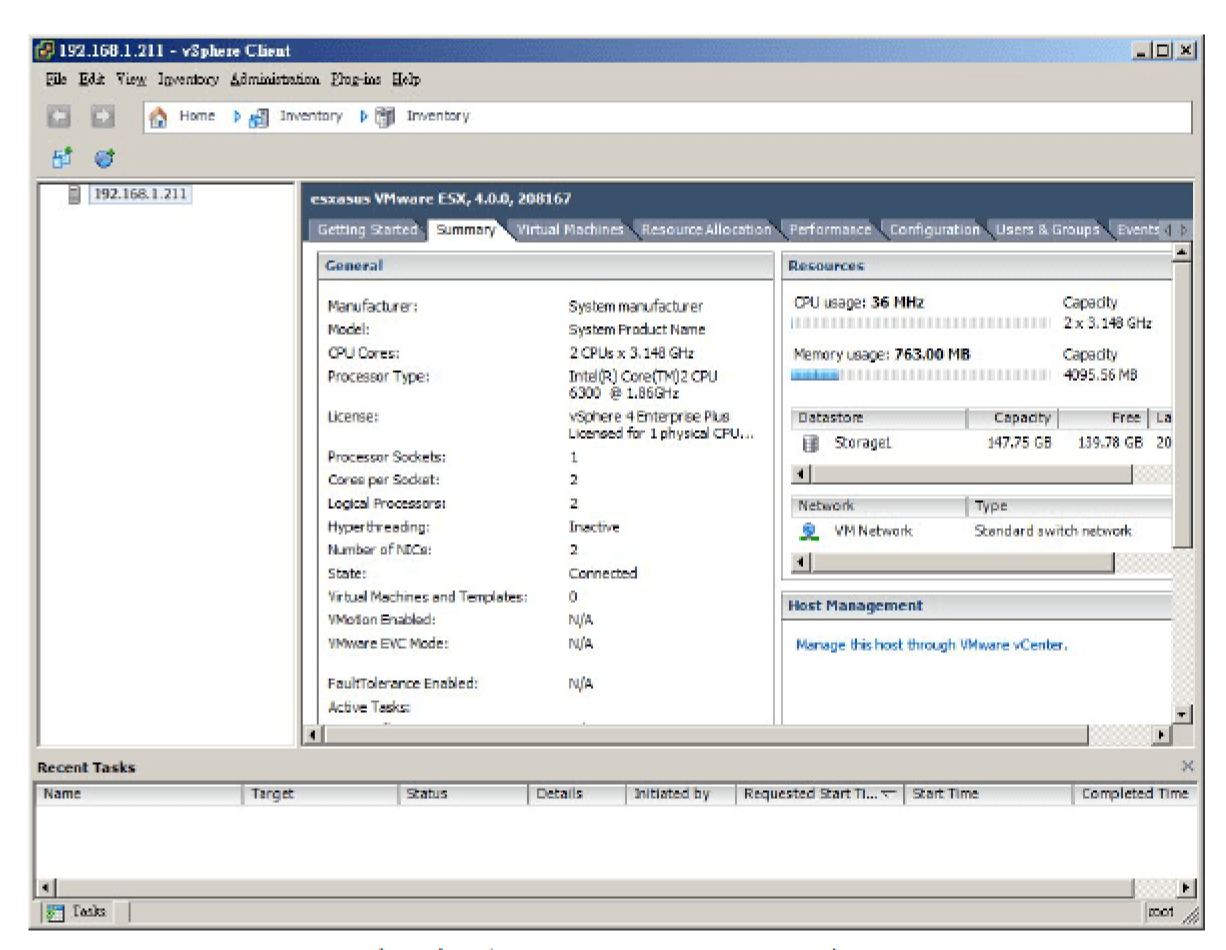
（4）配置虚拟机中的 BIOS。

（5）开始安装虚拟机。

（6）校调虚拟机中的操作系统（安装 VMware Tools）。

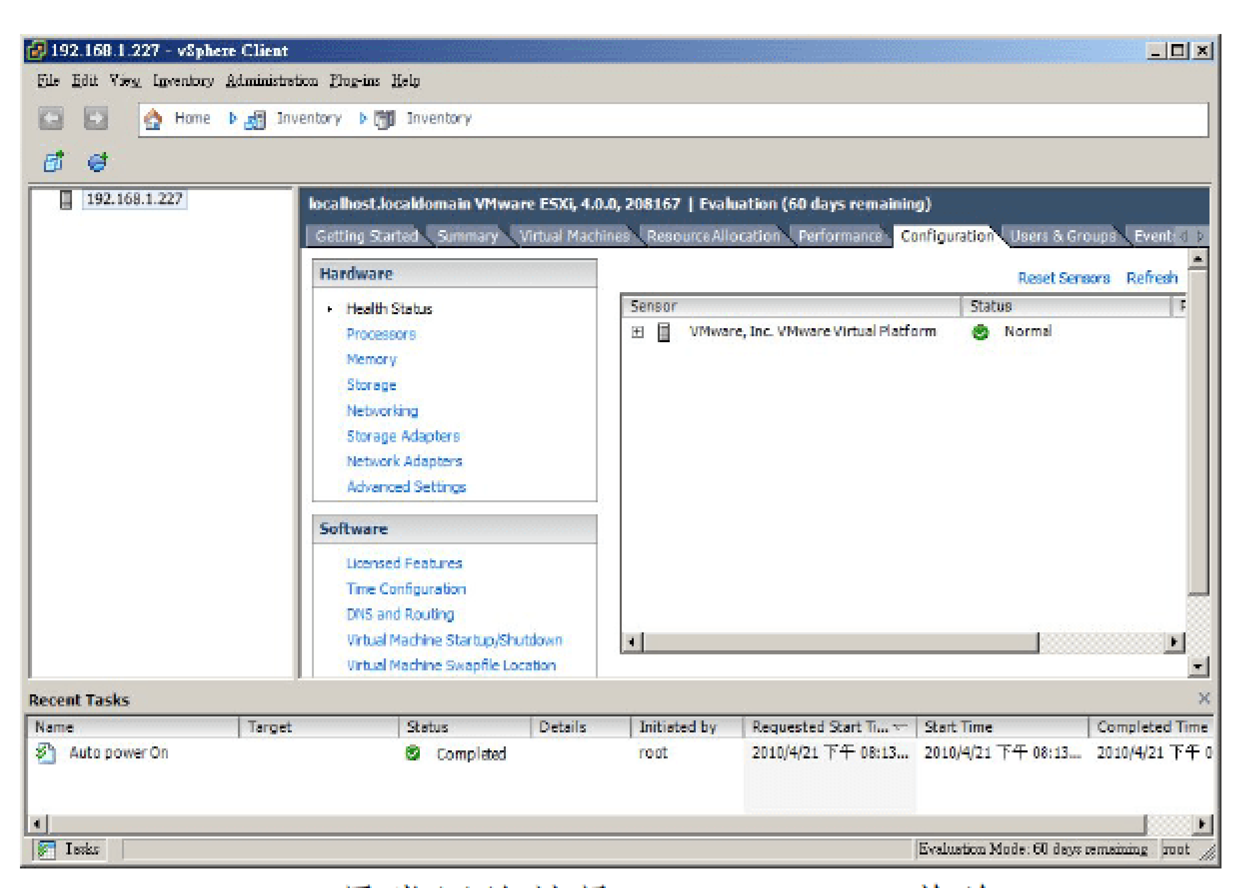
1. 理解 vSphere Client 的基本操作界面

在进入 vSphere Client 时，首先会弹出如下图所示的画面。我们会看到左边的窗格是当前的主机名，当有了虚拟机之后，就会有虚拟机在窗格下弹出。右边则是操作的菜单。我们会在下一章详细说明这些菜单的用处。



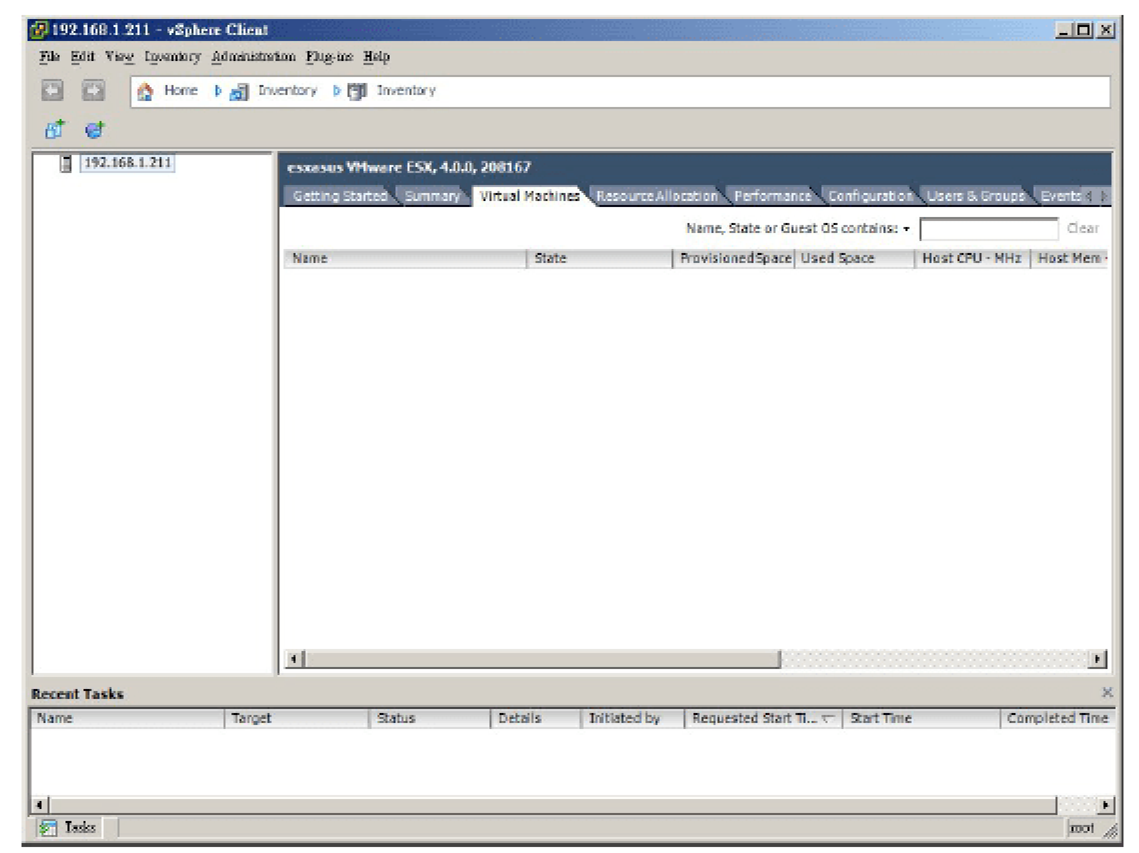
▲ 标准的 vSphere Client 主画面

一般在 vSphere Client 中，最常用的菜单就是 Configuration。这里我们可以配置各种不同的整体配置，如处理器、内存网卡和存储设备等。



▲ 最常用的就是 Configuration 菜单

另外一个常用菜单就是 Virtual Machines。所有和虚拟机有关的创建和配置也都在这个选项卡中。我们就从直接在 ESX 中创建一个虚拟机开始理解这些最常见的菜单。



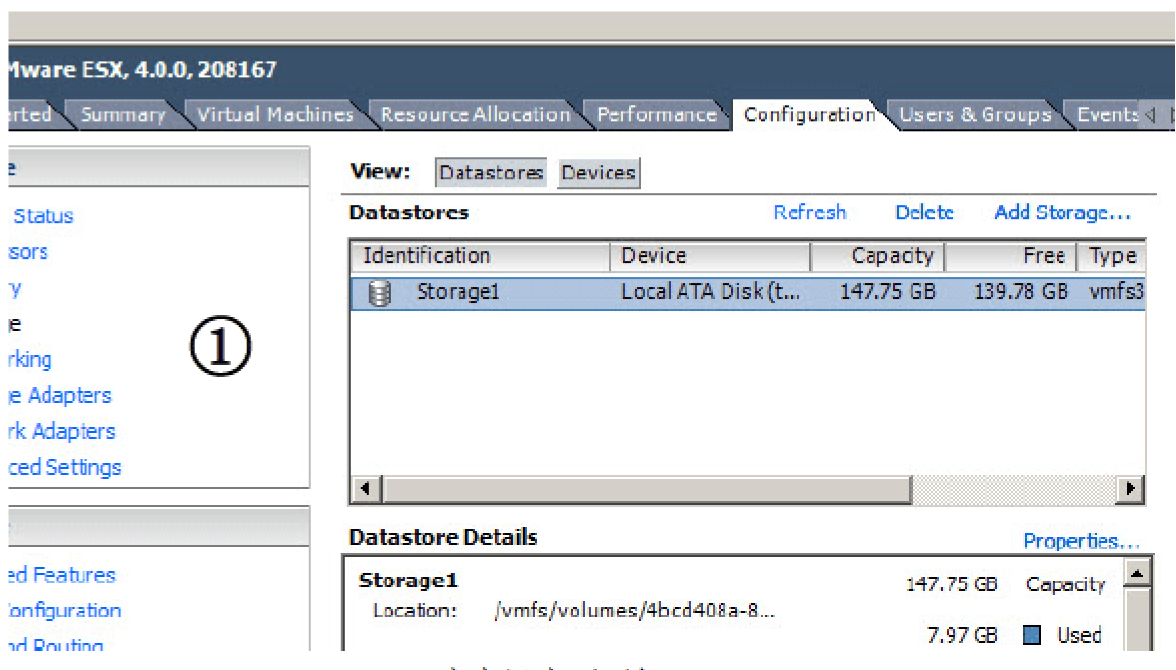
▲ 创建和针对单个虚拟机的操作则是在这个选项卡之下

2. 创建一个文件夹来存放未来的虚拟机

在 ESX 中，存放数据的地方称为 Datastore。由于我们是将 ESX 安装在本地硬盘，并且使用本地硬盘作为基本的虚拟机存放位置，因此本地硬盘的 VMFS 系统会被视为第一个默认的 Datastore。我们就将第一台虚拟机的文件夹创建在这个默认的 Datastore 中。

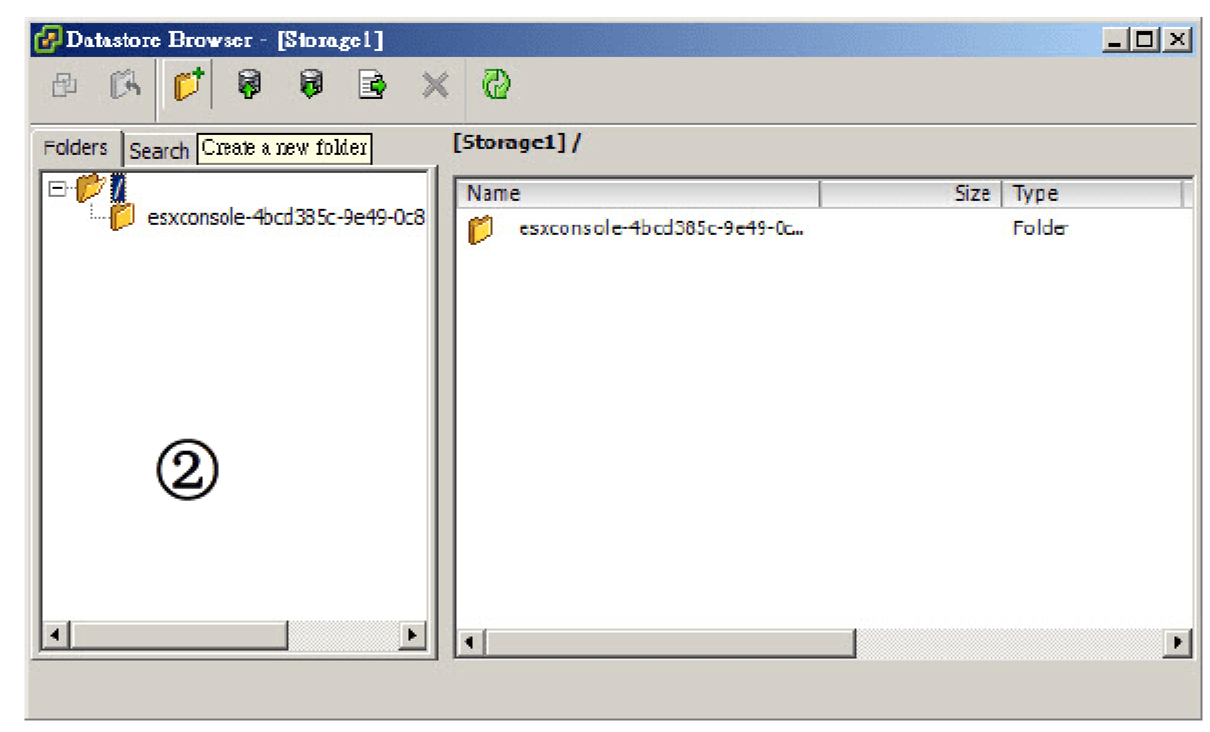
▶ 创建虚拟机存放的文件夹

1．在 vSphere Client 中，选择 Configuration/Storage，并且双击 Storage1 这个 Datastore。



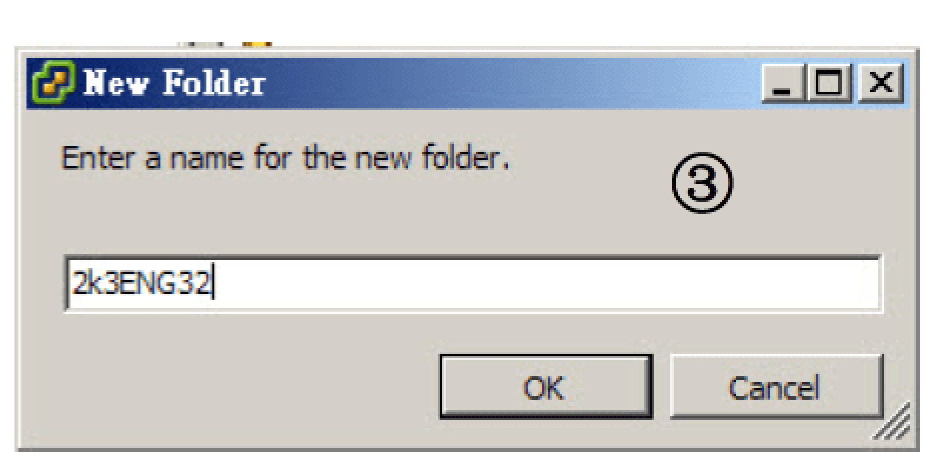
▲ 选择默认的 Datastore

2．之后会弹出这个文件夹的结构。我们可以单击上方的创建文件夹的图示来创建新的文件夹。

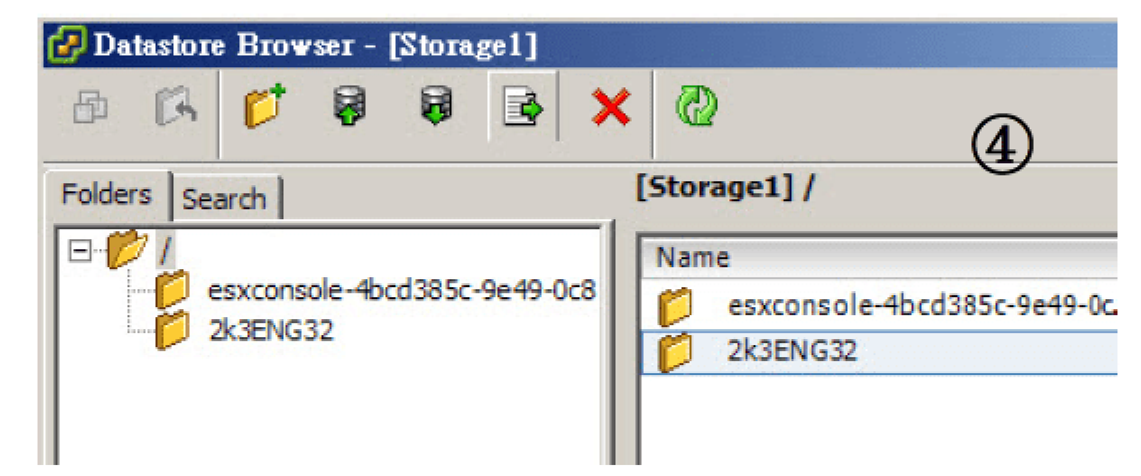


▲ 单击图示来创建新的数据

3. 此时我们键入文件夹名称，如 2K3ENG32，单击 OK 按钮。
4. 之后你可以看到这个创建好的文件夹。在创建虚拟机时，虚拟机的名称会和文件夹一样，换句话说，你的虚拟机会自动存放在对应名称的文件夹中。
5. 我们也创建一个新的文件夹来存放以后的 ISO 文件。

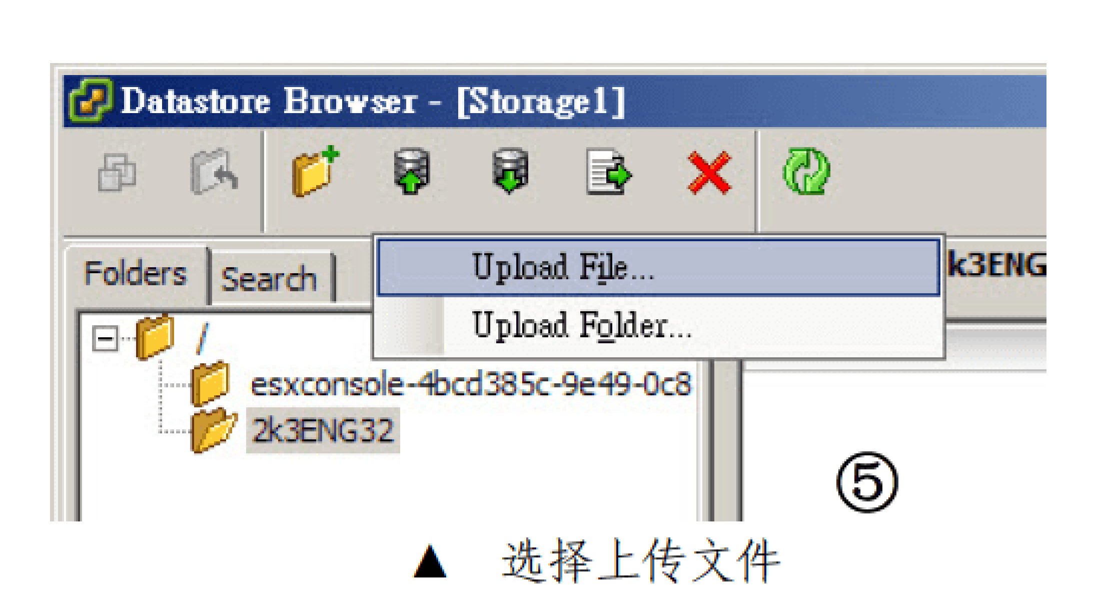


▲ 创建新文件夹名称 2K3ENG32，要和虚拟机的名称一样

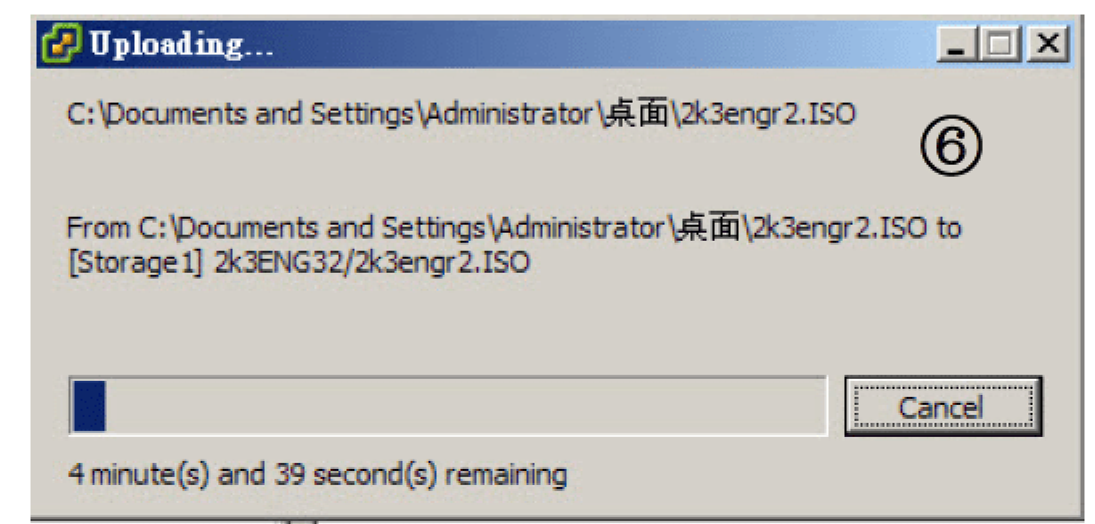


▲ 创建一个虚拟机和一个 ISO 档的文件夹

6. 选择从本地上传到 ESX 的图示，这里选择 Upload File。
7. 从本地选择你要上传的 ISO 档，我们这里就以 Windows Server 2003 的安装 ISO 文件为例子。

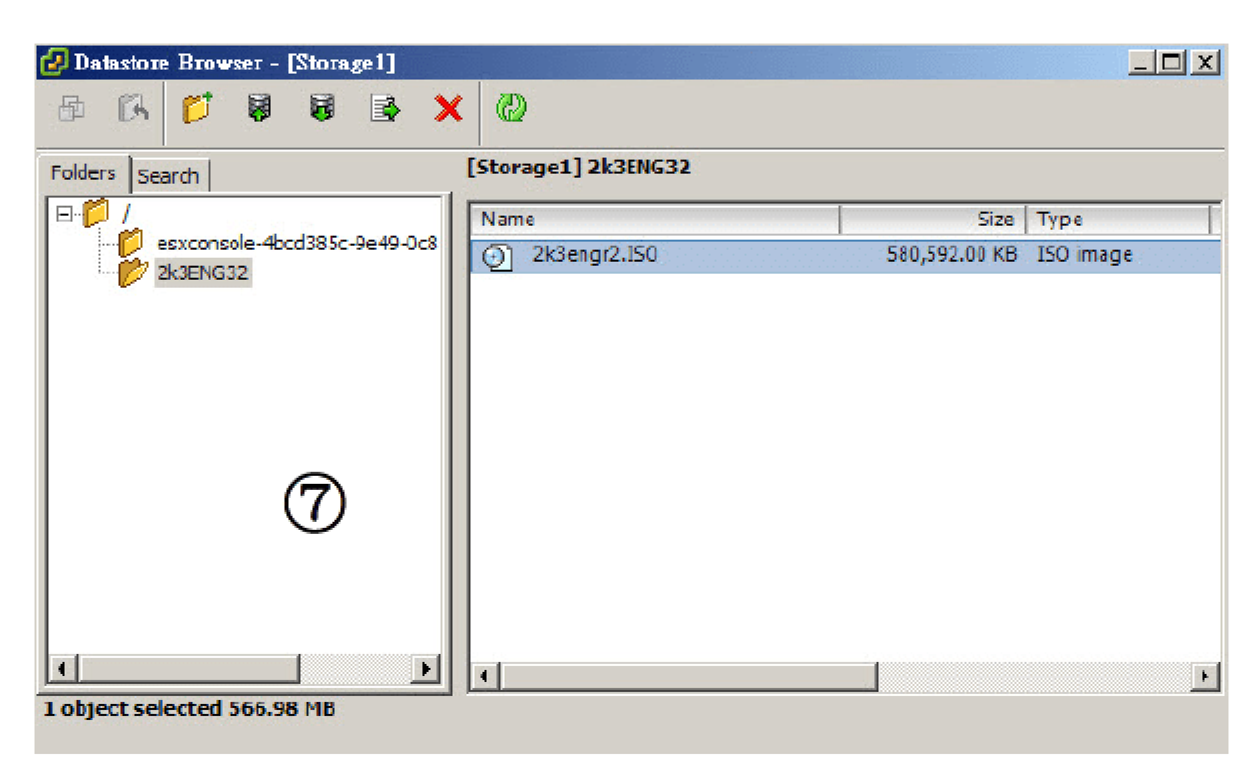


▲ 选择上传文件



▲ 上传 Windows Server 2003 的 ISO 档

8. 选择完之后会弹出报警消息，我们不需要理会这个消息，单击 Yes 按钮直接上传。
9. 此时会进入上传窗口，当传完时就可以在 Datastore 中看到上传的 ISO 档了。



▲ ISO 已经上传完毕了

注意

在 ESX Server 文件结构中真正的存放位置

虚拟机在 vSphere Client 中配置的位置，在 ESX 中是有固化的存放位置的。一般在 ESX 根目录下的/vmfs/volumes 中。如果你是使用默认值来安装 ESX，/vmfs 事实上是挂载在 ESX 下的一个分区/dev/sda3。知道这个位置的用处是，你可以从 Linux 的命令行来提交 ISO 文件到这个位置，以便安装虚拟机中的操作系统。当然我们也可以通过 vSphere Client 中的命令来提交 ISO 文件。

```
login as: root
root@192.168.1.88's password:
Last login: Wed Apr 21 12:01:29 2010 from mainwork
[root@ESXonWK7 ~]#
[root@ESXonWK7 ~]#
[root@ESXonWK7 ~]# cd /vmfs
[root@ESXonWK7 vmfs]# ls
devices  volumes
[root@ESXonWK7 vmfs]# cd volumes/
[root@ESXonWK7 volumes]# ls -al
total 1024
drwxr-xr-x 1 root root  512 Apr 21 13:17 .
drwxrwxrwt 1 root root  512 Apr 21 11:22 ..
drwxr-xr-t 1 root root 1120 Apr 19 13:06 4bcbe4e3-a0e6b5a0-b44f-000c29c34ec2
lrwxr-xr-x 1 root root   35 Apr 21 13:17 Storage1 -> 4bcbe4e3-a0e6b5a0-b44f-000c
29c34ec2
[root@ESXonWK7 volumes]#
```

▲　一般来说会存放在/vmfs/volumes 之下

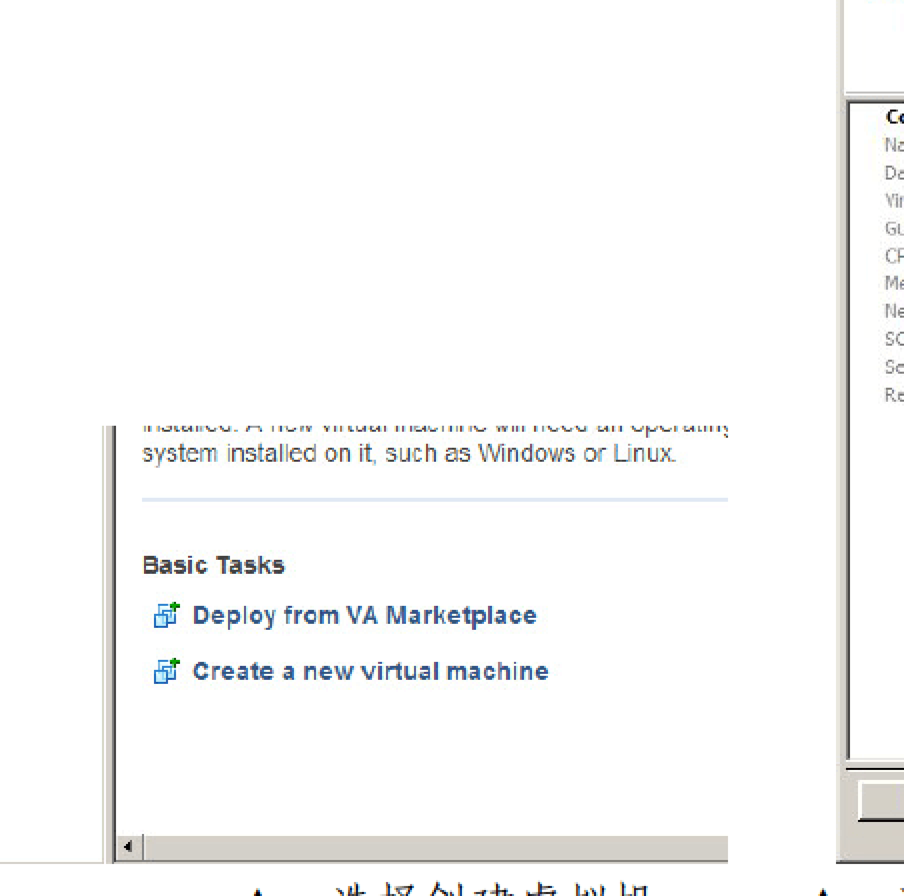
3. 创建 ESX 下的虚拟机

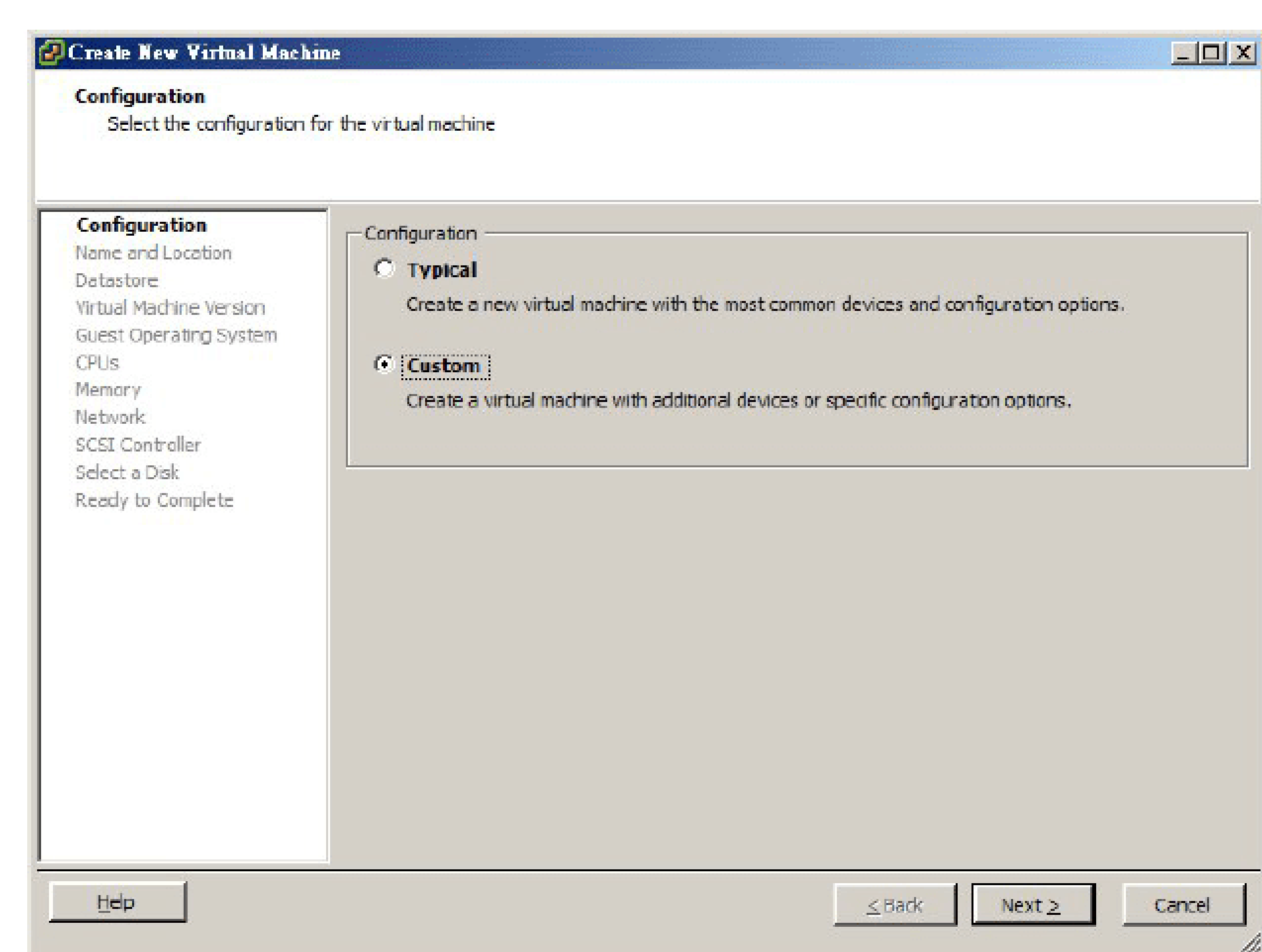
如果你有使用过 VMware Workstation 或 VMware Server 的经验，你会发现 ESX 的界面和这两个虚拟机产品几乎一模一样，尤其是 VMware Server。在 ESX 上创建虚拟机和 VMware Server 的操作一样，而观念则是和 VMware Workstation 很贴近。但 ESX 作为一个企业级的虚拟机产品，在配置上是更加丰富的。我们就试着创建一个 Windows 2003 的虚拟机。

▶ 在 ESX 上创建第一台虚拟机

1. 进入 vSphere Client，并且选择 Create a new virtual machine 选项。

2. 此时会进入配置模式，单击 Custom 单选按钮，之后单击 Next 按钮继续。





▲　选择创建虚拟机　　▲　选择自定义。ESX 和 VMware 的其他产品线起着共同的界面

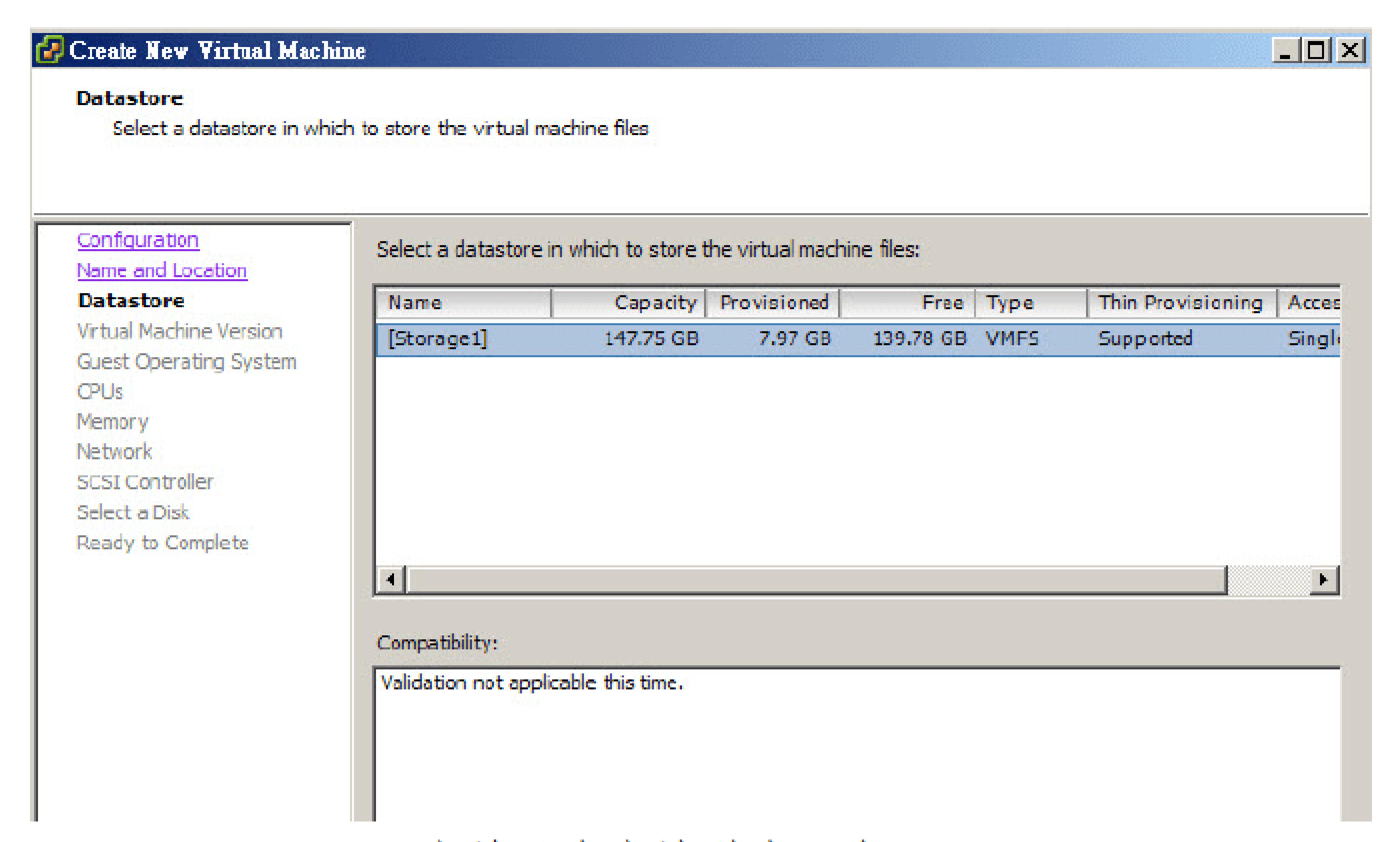
3. 之后键入虚拟机的名称。记住要和之前创建的目录名称一样，虚拟机才会压入该目录。

之后单击 Next 按钮继续。



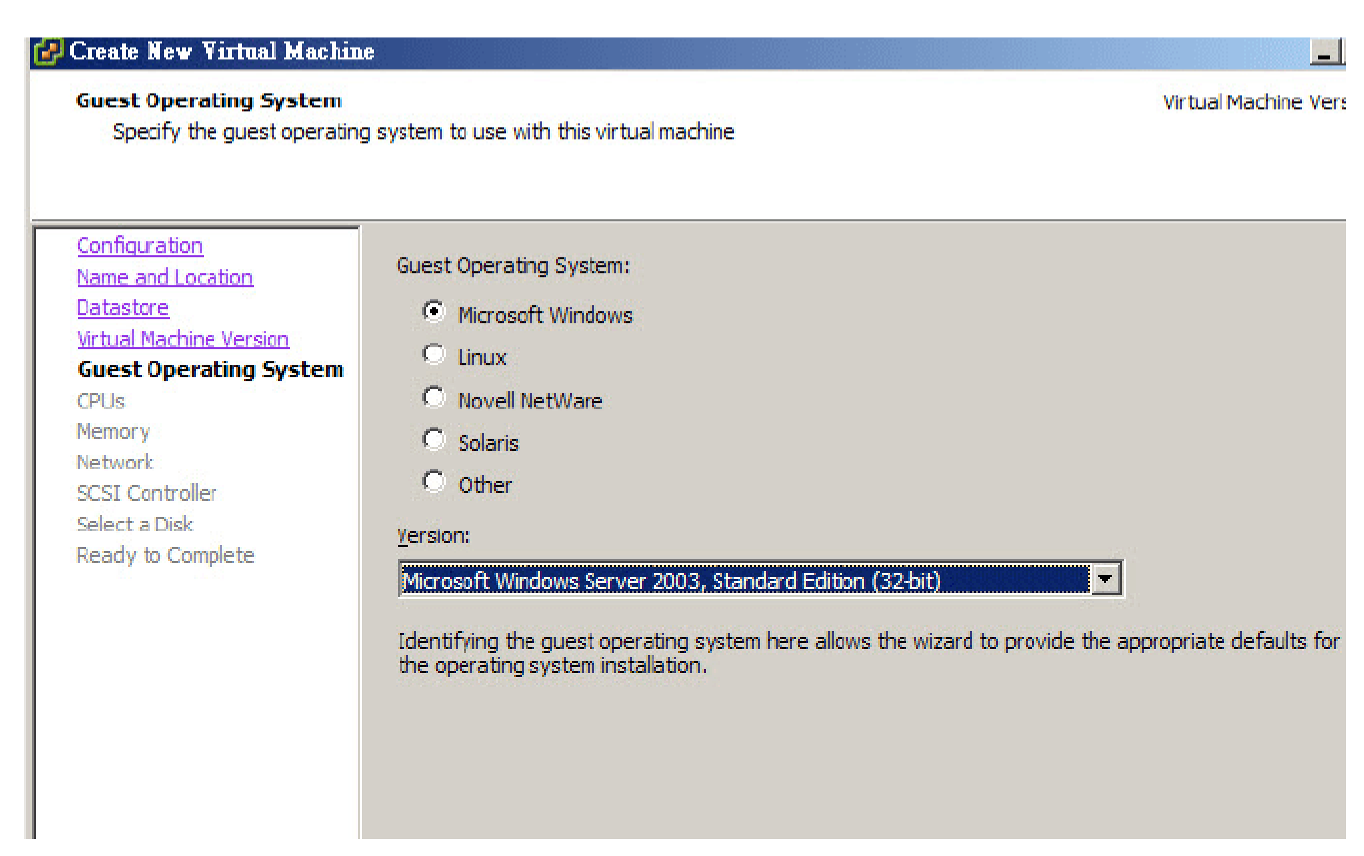
▲ 这里的虚拟机名称和文件夹名称要一样才会放入该文件夹

4. 接下来是选择存储的地点，当前是选择唯一的 Datastore 本地硬盘，单击 Next 按钮继续。



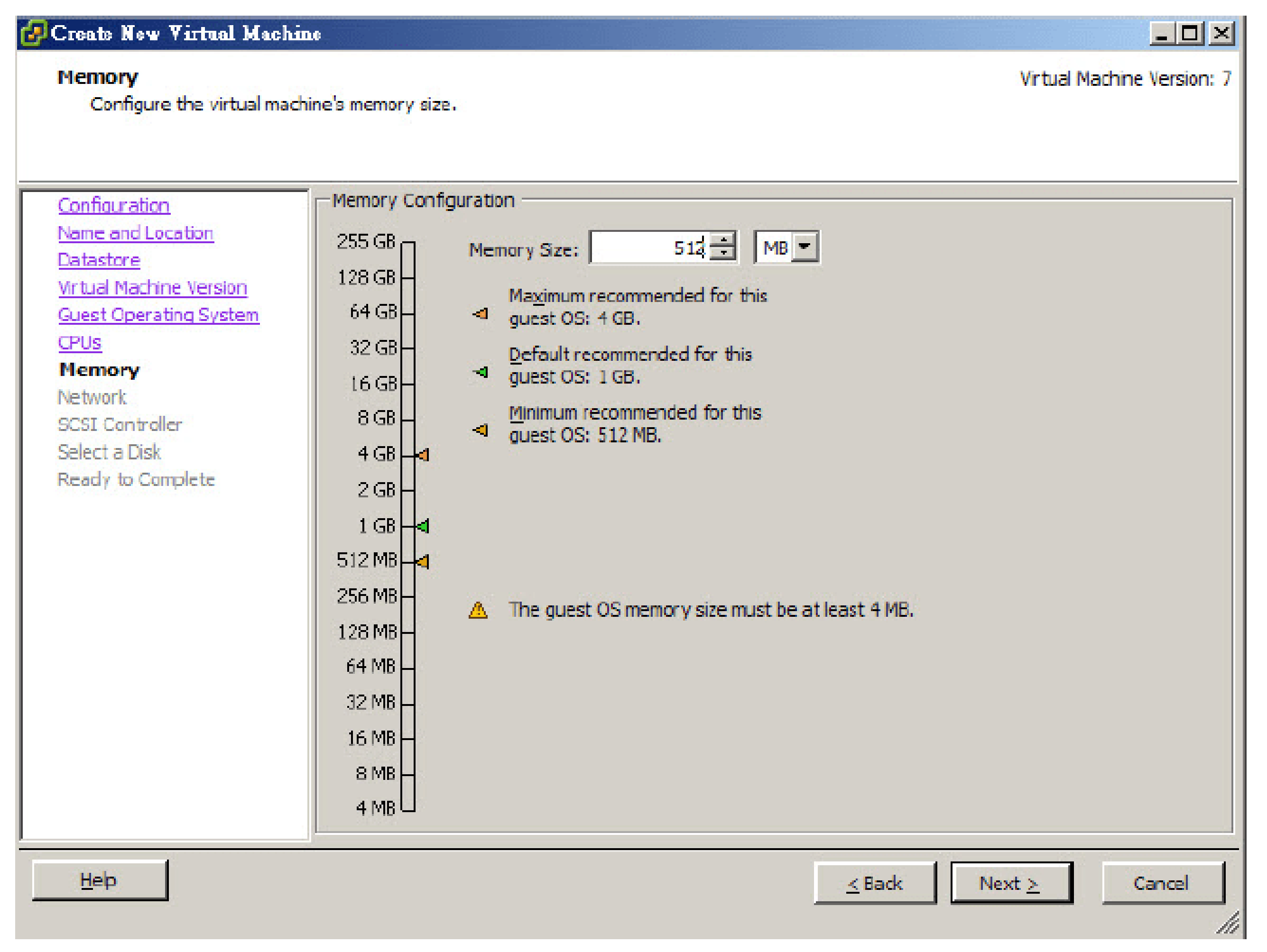
▲ 当前只有本地磁盘一个 Datastore

5. 接下来会先选择兼容性，我们可以任选一个。然后是选择操作系统。我们在此选择 Windows Server 2003，因为在 ESX 上安装 XP 需要额外的驱动程序。单击 Next 按钮继续。



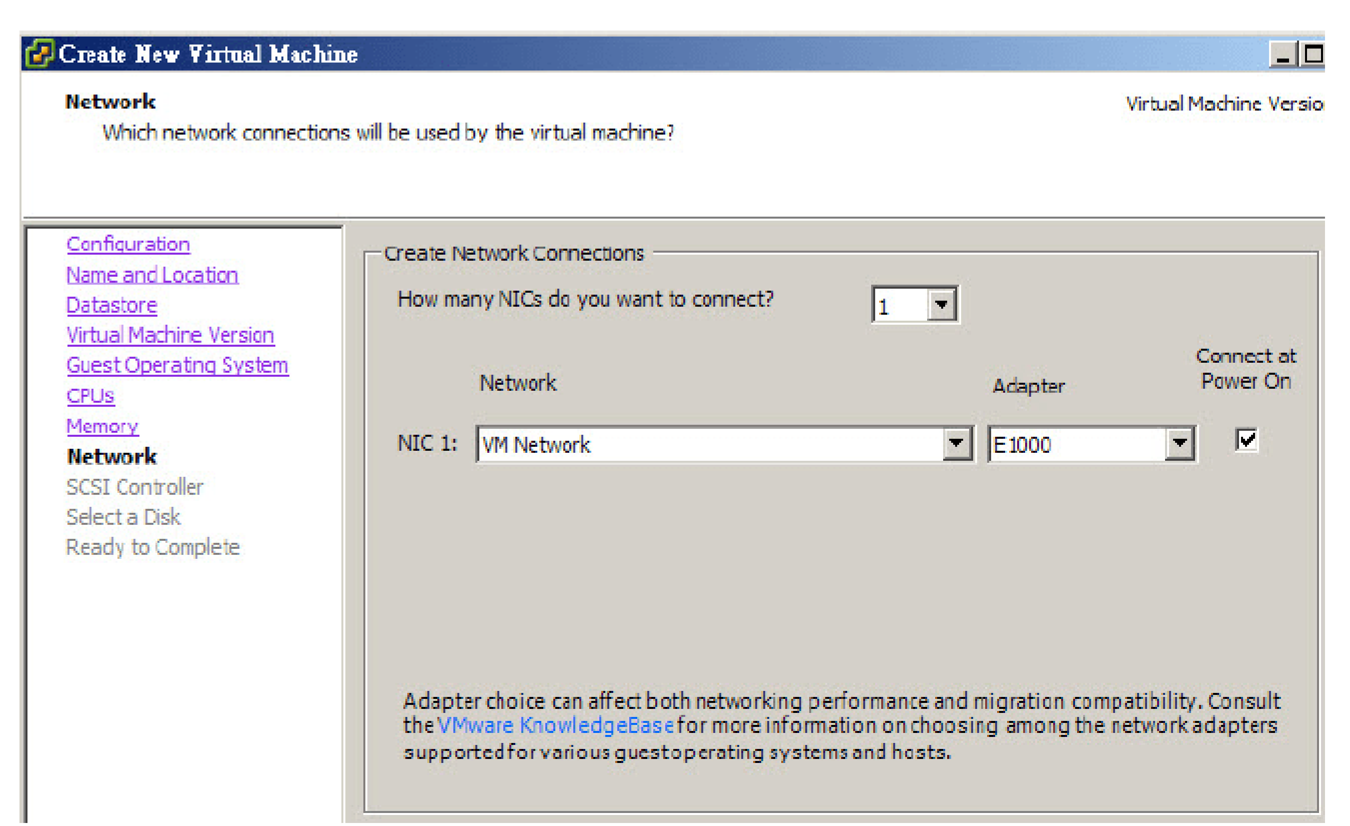
▲ 选择安装 Windows Server 2003

6. 接下来是配置 CPU 的数量和内存的大小，这里和任何虚拟机产品的方式一样。给 Windows Server 2003 1 个 CPU、512MB 的内存，单击 Next 按钮继续。



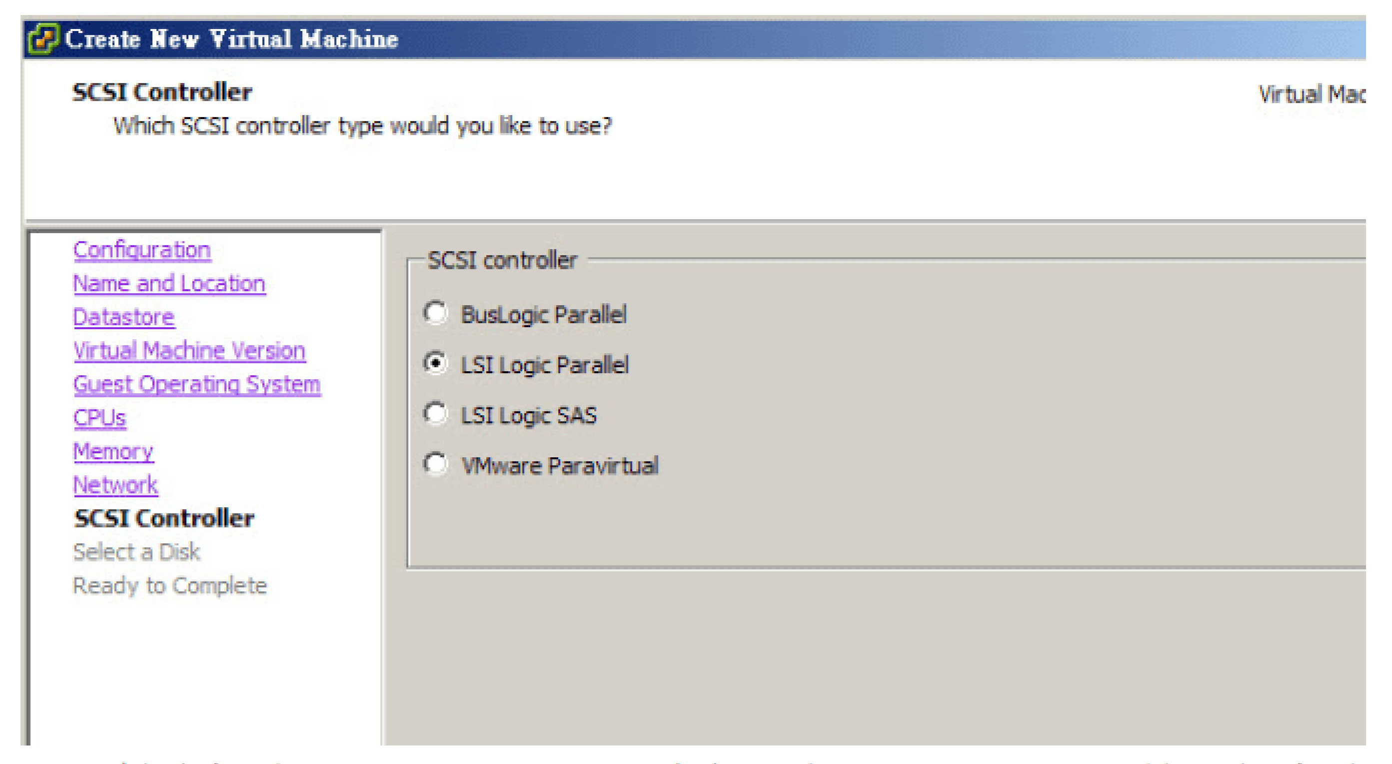
▲ 配置 512MB 的内存

7．接下来是网卡的配置。在 ESX 下，网卡的配置和 VMware Workstation 是完全不同的，在下一章会有详细说明。这里就先选择默认值，单击 Next 按钮继续。



▲ 在 ESX 中的网卡选项较少，主要是看 ESX 本身的网络连接方式

8．接下来是创建虚拟硬盘，选择创建新的硬盘，单击 Next 按钮继续。



▲ 创建新的 VMware VMDK 磁盘。这和 Workstation 是不相容的

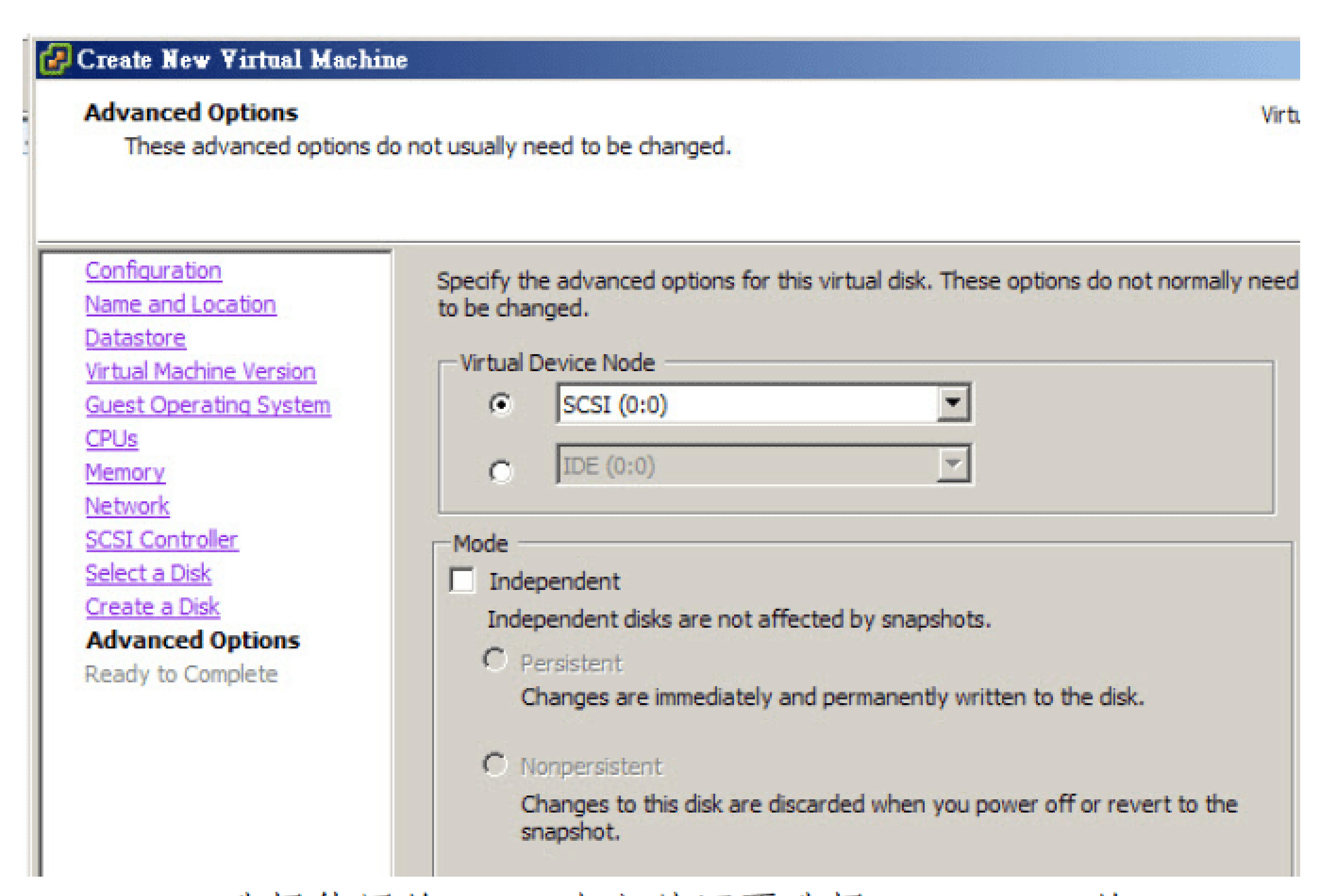
9．Windows Server 2003 占用的空间不大，选择 8GB 就足够了。其他的配置不需要更改，

会将虚拟硬盘存放在同一个目录下。单击 Next 按钮继续。

Create New Virtual Machine
Create a Disk
Specify the virtual disk size and provisioning policy
Configuration
Name and Location
Datastore
Virtual Machine Version
Guest Operating System
CPUs
Memory
Network
SCSI Controller
Select a Disk
Create a Disk
Advanced Options
Ready to Complete
Capacity
Disk Size: GB
Disk Provisioning
Allocate and commit space on demand (Thin Provisioning)
Support clustering features such as Fault Tolerance
Location
Store with the virtual machine
Specify a datastore:
Browse...

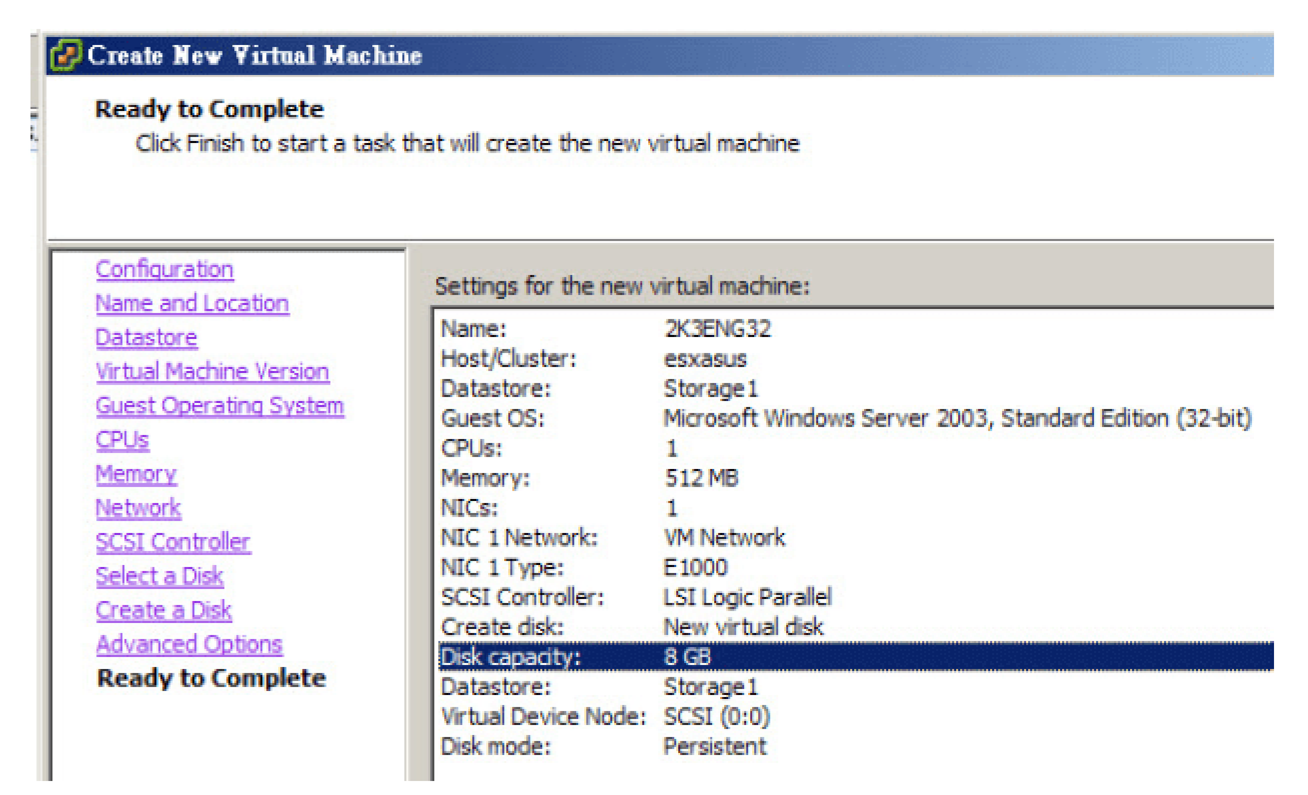
▲ 给定虚拟硬盘的大小

10．之后选择硬盘的界面，ESX 提供了 SCSI 界面，因此我们要给定 SCSI 的编号，使用默认值即可。单击 Next 按钮继续。



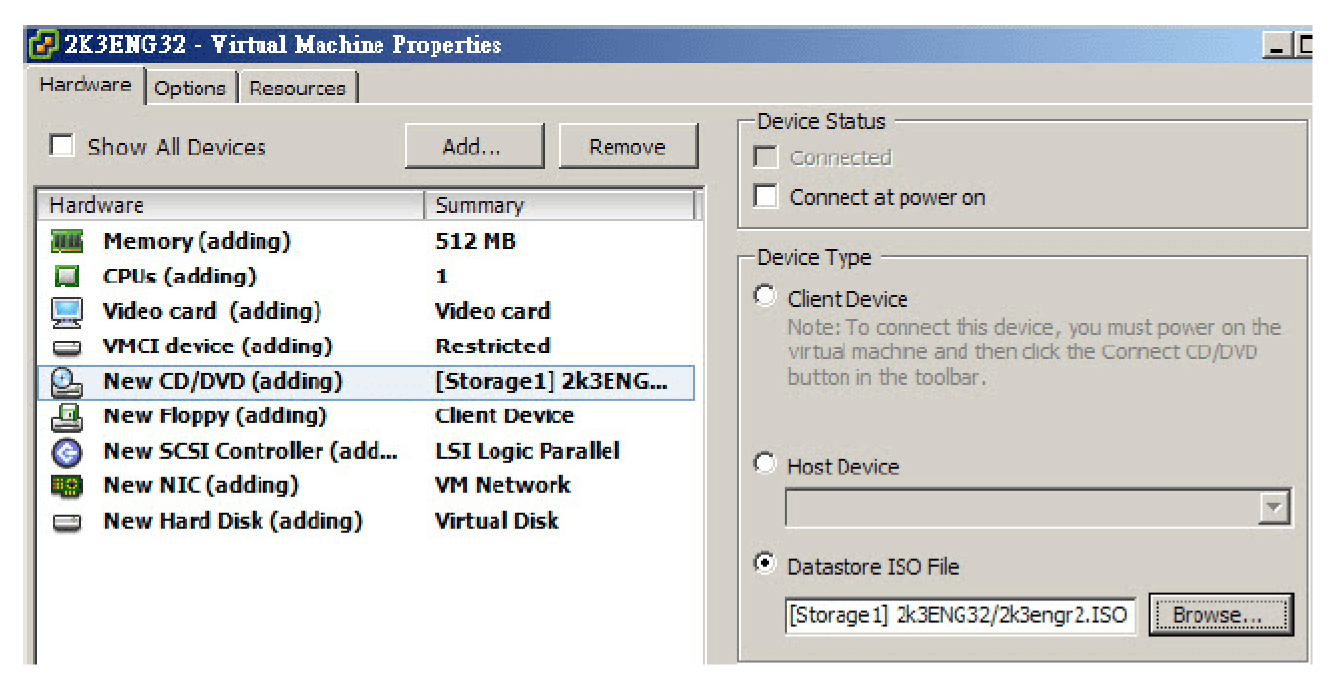
▲ 选择使用的 SCSI 卡之外还要选择 SCSI LUN 的 ID

11．当弹出下图所示画面时，意味着虚拟机已经准备完毕了。



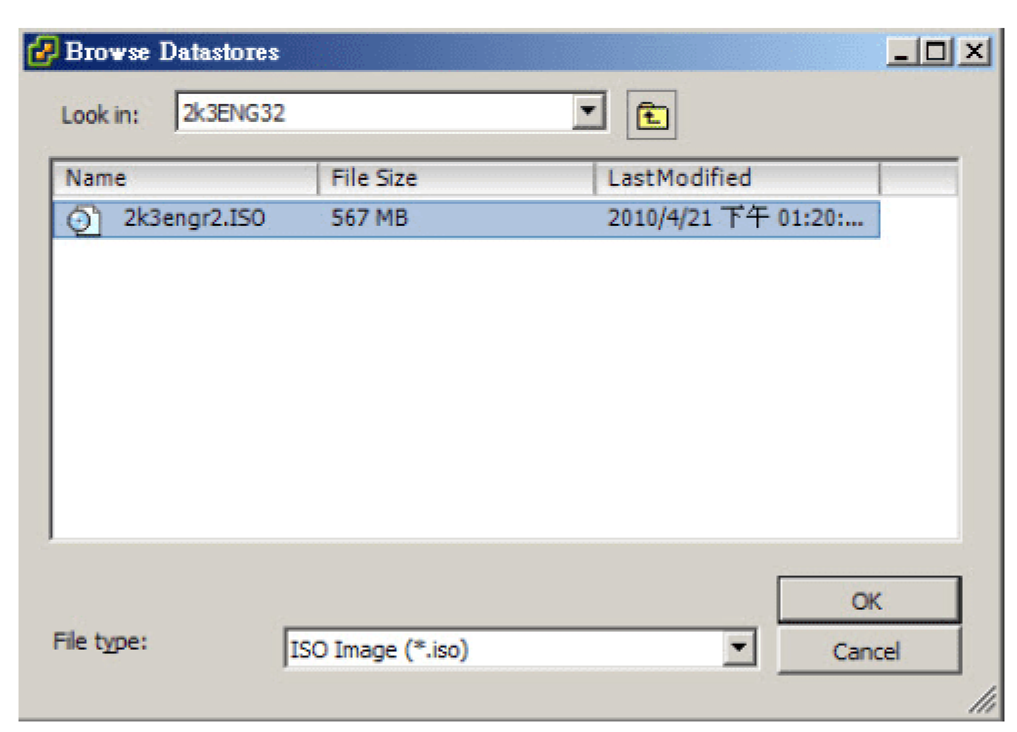
▲ 配置完毕了

12．此时会进入虚拟机的配置画面。选择 CD/DVD 选项，并且给定这个 CD/DVD 要挂载到 ISO。



▲ 将刚才上传的 ISO 挂载至这台虚拟机的光驱上

13．选择刚才上传的 ISO 文件。



▲ 刚才都放在 ISO 这个文件夹中了

当选择完毕之后单击 OK 按钮即可。

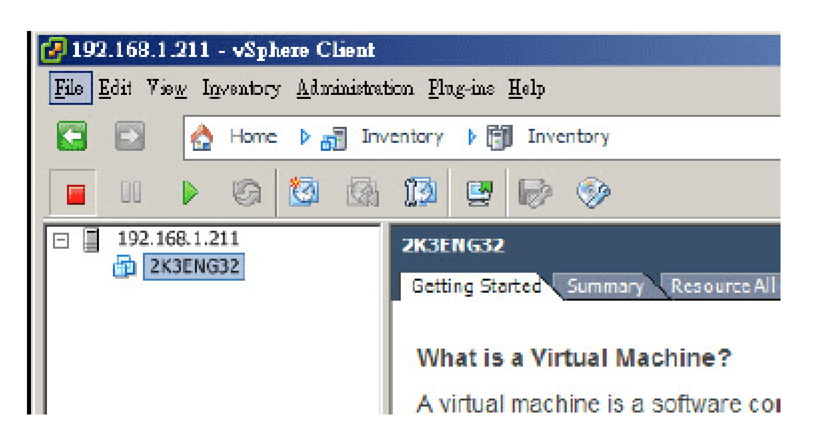
4．理解 ESX 中虚拟机的操作

在给定好虚拟机的光驱之后，你的第一台 ESX 虚拟机已经准备好了。接下来就是安装操作系统。当虚拟机安装好之后，整个 vSphere Client 的界面就会发生改变。首先你可以在左边的窗格看到新建的虚拟机。当你将游标移到这个新建的虚拟机时，右边的菜单也会改变，这也是我们操作虚拟机的地方。

▶ 虚拟机任务台的基本操作

- 上方工具条：有关虚拟机的操作都在这里，这些图示也很直白。如开机关机、挂起、重新激活等等。还有一个就是激活单独的虚拟机总控 （即虚拟机的桌面），这是非常实用的。
- Getting Started：这是虚拟机的基本操作，如激活、挂起或编辑参数。
- Summary：当前这台虚拟机在整个 ESX 中的状态，如 CPU 数、内存数量、使用的 Datastore 和网络参数。
- Performance：当前这台虚拟机的 CPU 占用率以及历史占用率等。
- Events：当前这台虚拟机所发生的任何表项，这也是虚拟机的 LOG 档。

- Console：这是你最常用的虚拟机总控台，也是这台虚拟机的桌面。
- Permissions：这是使用这台虚拟机的权限，在多人使用环境中，这个配置相当重要，可以决定谁能总控这台虚拟机。

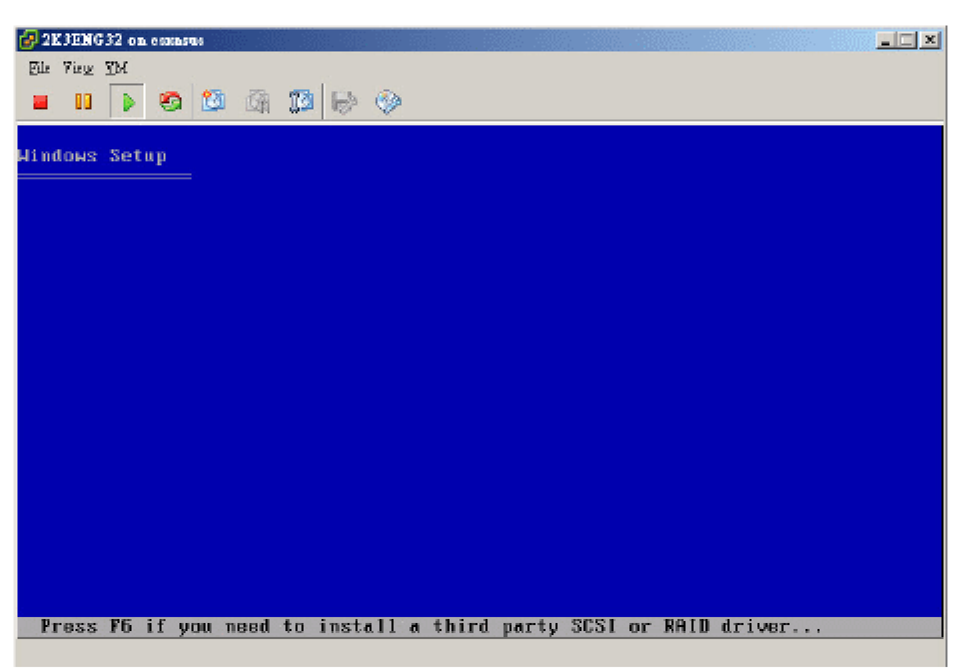


▲ 在有虚拟机的情况下，菜单中的命令也不同，图中为引导的按钮

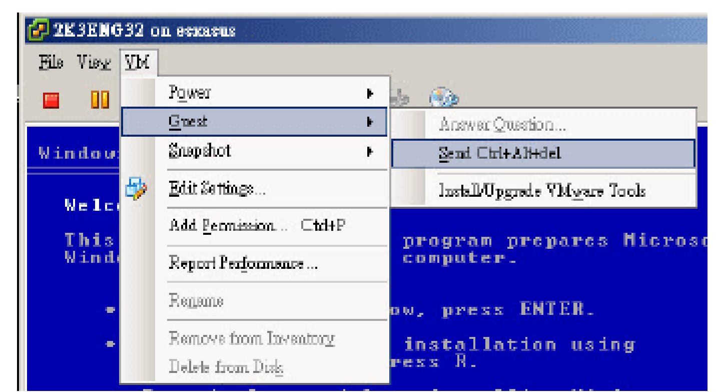
> **小常识**
>
> **进入和离开虚拟机桌面**
>
> 当我们进入 Console 时，如果想进入虚拟机的主控台，也就是这台计算机的“桌面”，只需要在这个桌面上的任何一个地方单击鼠标即可。如果你想离开虚拟机而回到“真实世界”的 vSphere Client 时，只要按 Ctrl+Alt 组合键即可。在 VMware 的系列产品中，所有的离开虚拟机桌面的按键都是这两个。

5. 安装虚拟机的操作系统

在虚拟机中安装操作系统就像物理机一样，只要你将光盘给定好 ISO 档，并且将虚拟机的 BIOS 中配置光盘机引导即可。虚拟机的 BIOS 和物理机一样，在配置完毕之后，你就会发现虚拟机从光盘激活了。安装 Windows Server 2003 的过程就像在物理机中安装一样，我们在此就不详细说明了。



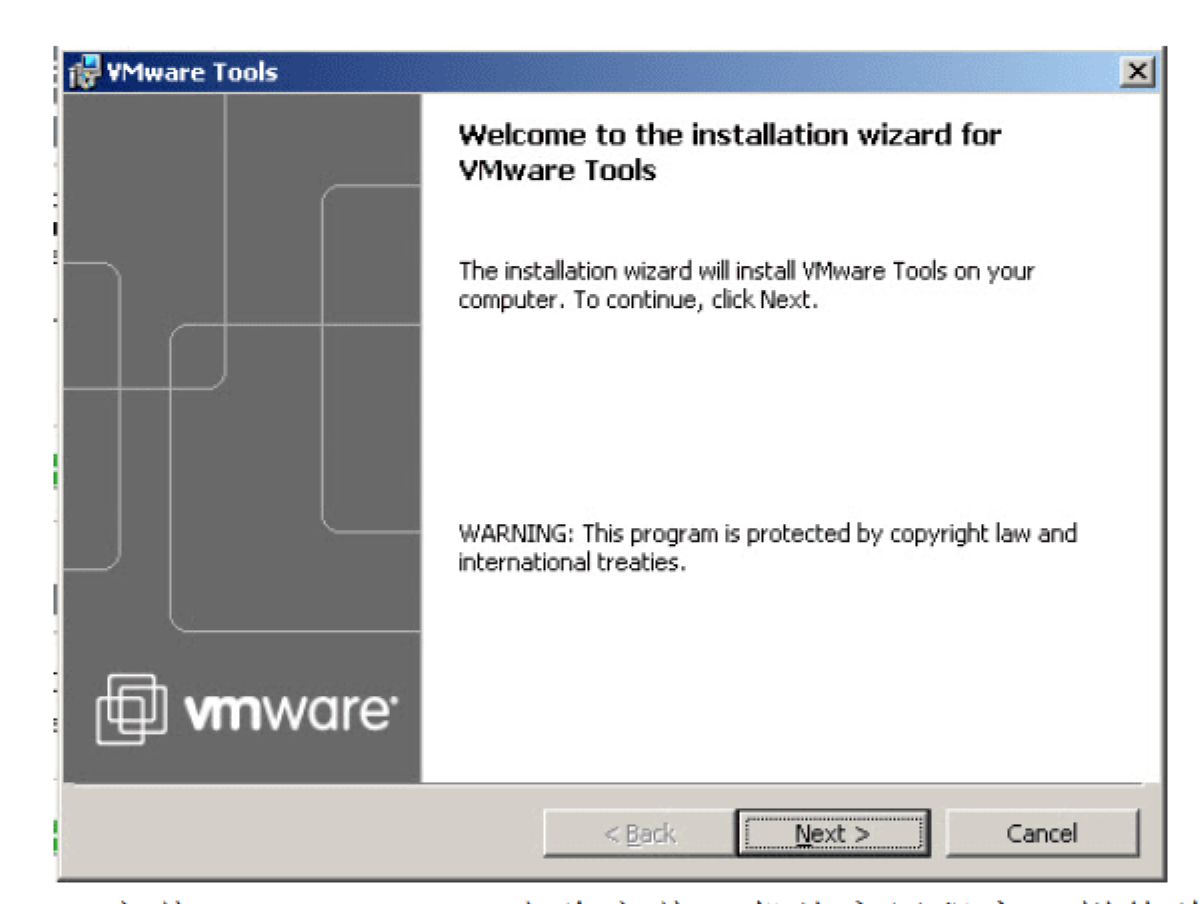
▲ 就像在物理机上安装一样

> **注意**
>
> **在虚拟机的桌面按下 Ctrl+Alt+Del**
>
> 许多操作系统在登录桌面时会要求用户按 Ctrl+Alt+Del 组合键。当你在虚拟机中，按下这三个组合键时，你会发现由物理机接收这个键的命令，组合键的作用并无法送入虚拟机中。此时我们可以在左边的虚拟机上右击选择送出这个组合键的命令。如果是在独立的虚拟机主控台，可以选择 VM 菜单中的“传送”组合键命令即可。
>
> 
>
>
> ▲ 要按下 Ctrl+Alt+Del 组合键必须选择菜单中的“传送”命令才行

6. 安装 VMware Tools

当安装好操作系统之后，千万要记得安装 VMware Tools。这个工具算是 VMware 在不同操作系统下的驱动程序。安装了 VMware Tools 之后，会在你的操作系统中安装显示和网卡的驱动程序，整个系统在虚拟机中的体现会变得比较顺畅，并且可以在虚拟机的主控台和 vSphere Client 的桌面之间移动鼠标，不需要按 Ctrl+Alt 组合键。VMware ESX 针对主流的操作系统都有提供 VMware Tools。

安装 VMware Tools 十分简单，只要在 vSphere Client 主控台虚拟机上右击，并且选择 Install/Upgrade VMware Tools 即可。系统会将当前使用的 DVD/CD 卸除，并且挂载上给定操作系统的内置 VMware Tools ISO 文件，你只要照着安装一般 Windows 实用程序的手段一步步运行即可，重启后，在右下角的状态栏看到 VMware Tools 的图示时就意味着安装完毕了。



▲ 安装 VMware Tools 也像安装一般的实用程序一样简单

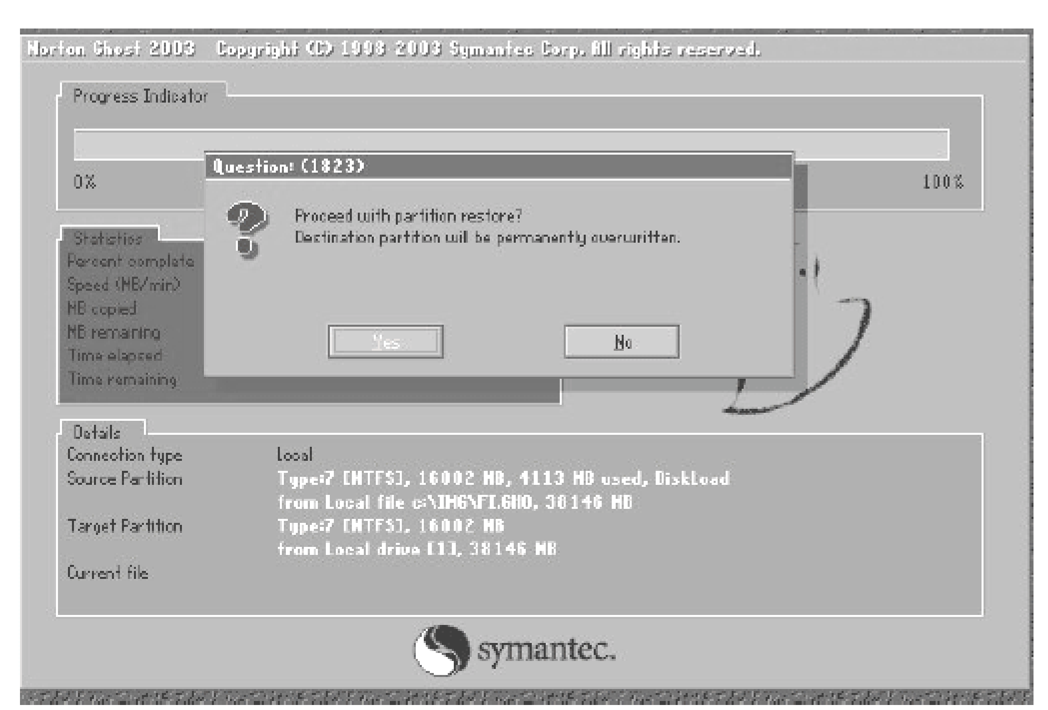
9.4 让 ESX/ESXi 更好用的技巧

ESX 中很多功能会让你的使用更加方便，其中最有名的就是快照了。快照是每一个虚拟机都有的基本功能，而 ESX 继承了 VMware 有名的快照管理。此外在 ESX 的使用上，我们也常常需要应用到一些第三方的软件来加强虚拟机的使用，我们在这一小节也会介绍。

9.4.1 理解 ESX 中的快照功能（Snapshot）

虚拟机除了在物理资源的利用率上大大超越物理机之外，最显著的优势就是管理上的功能。举例来说。如果你安装了 Windows Server 2008，并且为了某一个服务进行了配置。但在使用一阵子之后配置出现问题，这个服务就不能用了，这个时候怎么都找不出问题的答案时，最后只有重装一途了。

较聪明的做法就是在所有的配置都常规的时候先将整个硬盘 GHOST 起来，当弹出问题时，就将系统还原，但如果整个服务占用的空间很大，还原服务时系统的停机时间太长，用户将无法访问该服务。



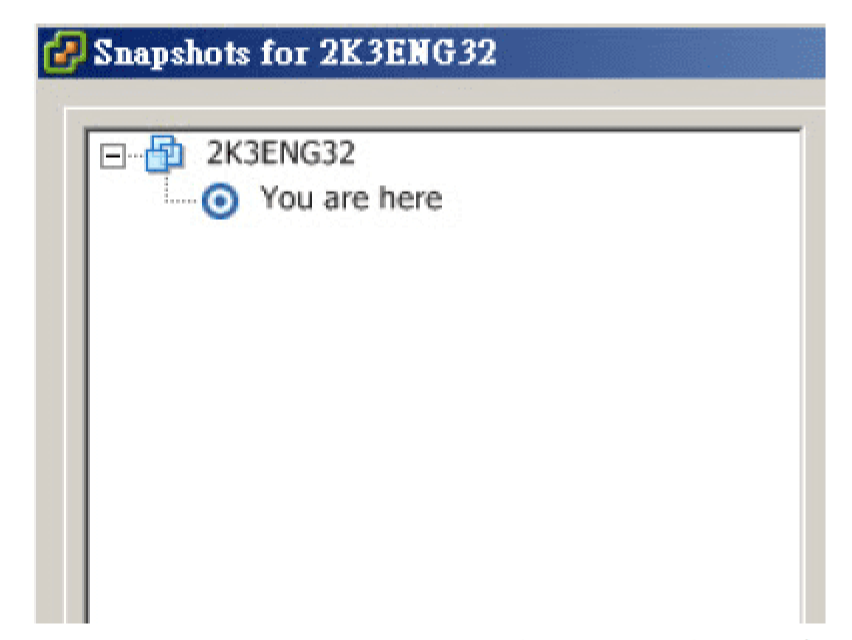
▲ 许多人用 GHOST 来解决系统蓝屏的问题，但相当耗时间

但如果使用的是虚拟机，还原这个服务可能只要几秒钟的时间，听起来貌似很神奇，但这就是虚拟机最大的优势之一。而要实现这个功能，只要简简单单应用 ESX/ESXi 的快照（Snapshot）功能即可办到。

1. 什么是快照？

大部分的虚拟机产品都有快照功能，包括微软的 Hyper-V 和本章的主角 VMware ESX/ESXi。快照主要是作为直线式的步骤性测试，举例来说，如果你需要安装 ABC 三套不同的软件，而这 ABC 必须照顺序安装，A→B→C。如果 A 没安装好，B 就不能安装上去，B 没安装好，C 就安装不上去，以此类推。如果我们没有快照，你可能安装好了 A，但是 B 一直安装不好，而重启又要重装，怎么办呢？

这时候我们可以先安装 A，然后在安装好 A 之后创建一个快照。接着开始安装 B。如果 B 没安装好，你就直接关机也行，因为你的快照 A 已经存下来了。接着一步步开始安装 B、C。每一次落实一个步骤就做一个快照。当你在安装到任何一个步骤发现有问题时，也可以"回退到上一个快照"。而所有的快照落实时，你也可以将 AB 两个快照都删除，只留下最后落实的 C。这种做法对于系统的还原和测试，比物理机的方便性的确强上很多。



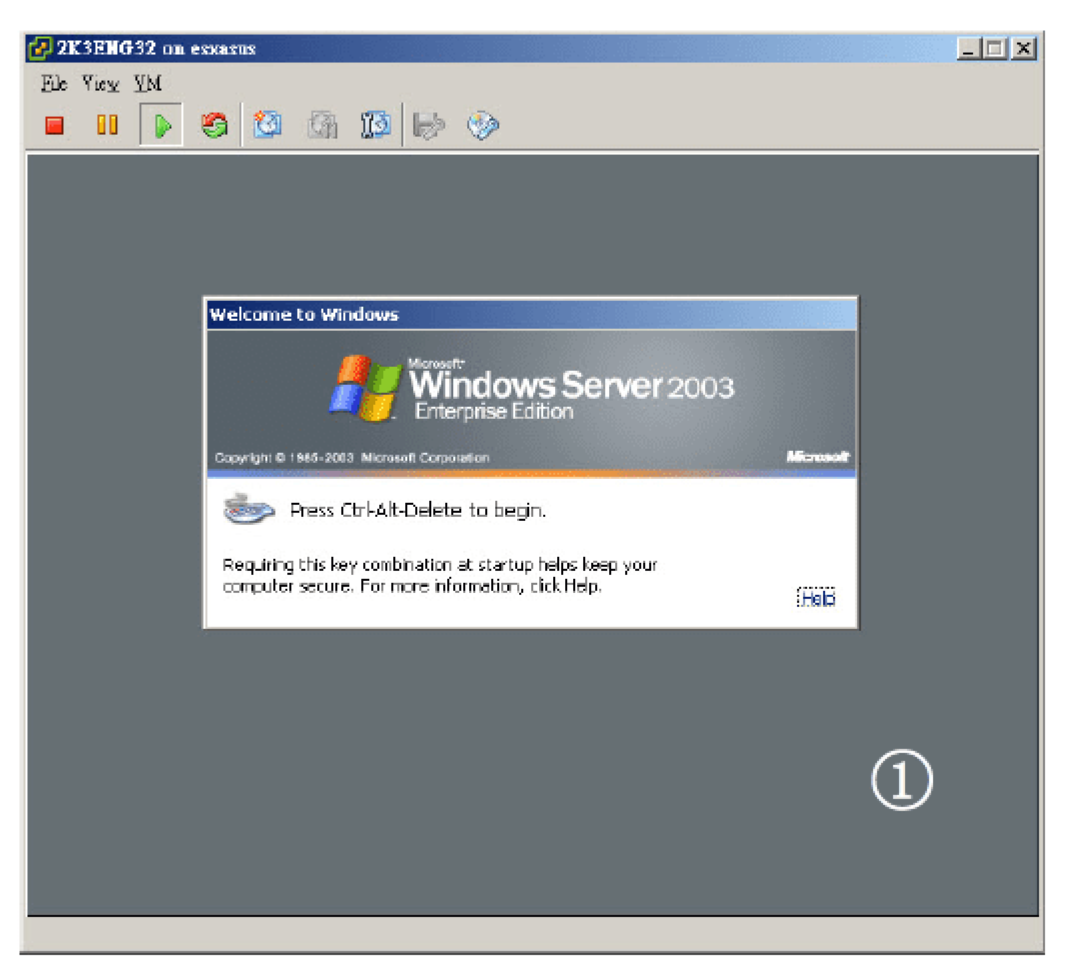
▲ VMware 产品线中最棒的就是快照功能

2. 使用快照功能

在 ESX 下替一个虚拟机制作快照时，可以是在引导或关机时，而创建快照之后，也可以导出，并且可以做出多个快照。ESX 和 VMware Workstation 一样，有个相当实用的快照管理员用来管理版本，我们就来看看下面的例子。

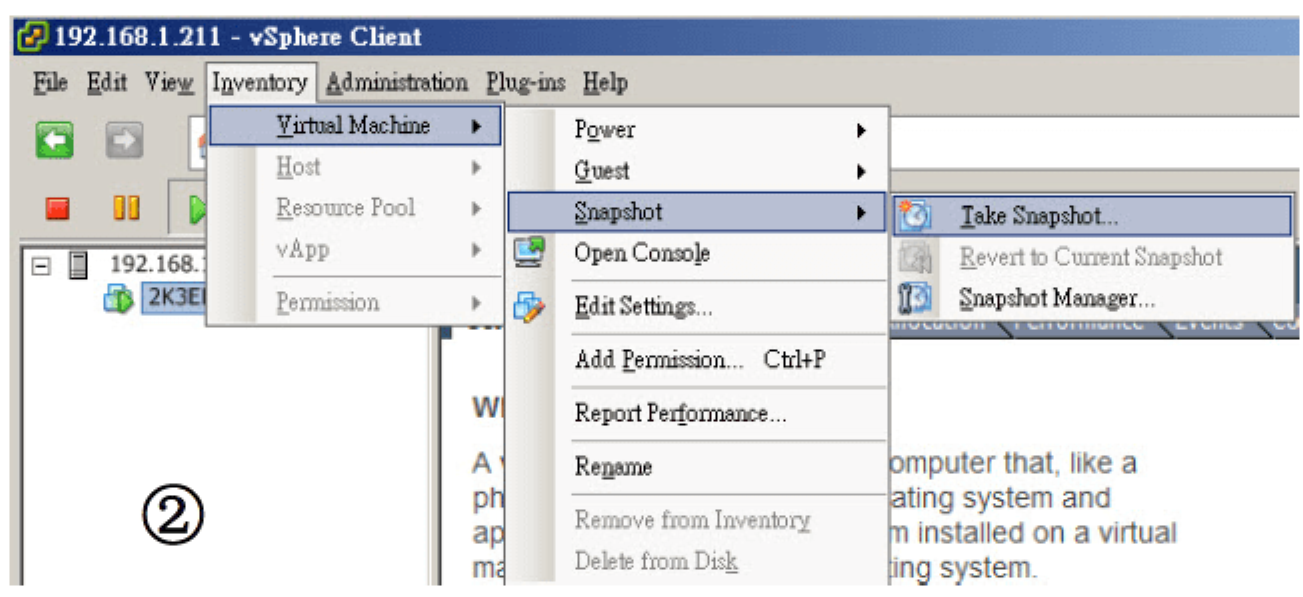
▶ 快照的创建

1. 如果我们有一个应用软件必须在 Windows 2003 上运行，但我们不知道这个软件能否在 SP2/SP3 上运行。因此我们安装一台 2003 系统来测试。



▲ 先安装刚才的 Windows Server 2003

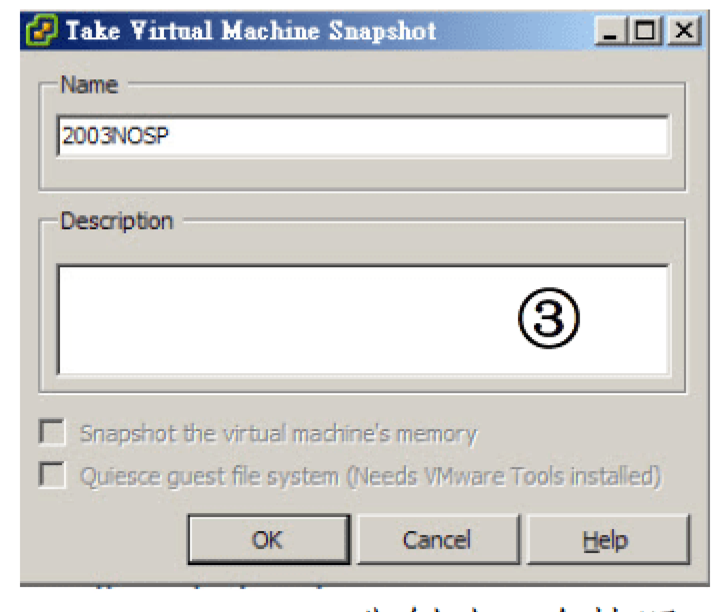
2．安装完毕之后，我们首先将这个完全没有安装任何 SP 的 2003 做一个快照。直接选择 Inventory/Virtual Machine/Snapshot/Take Snapshot 选项。



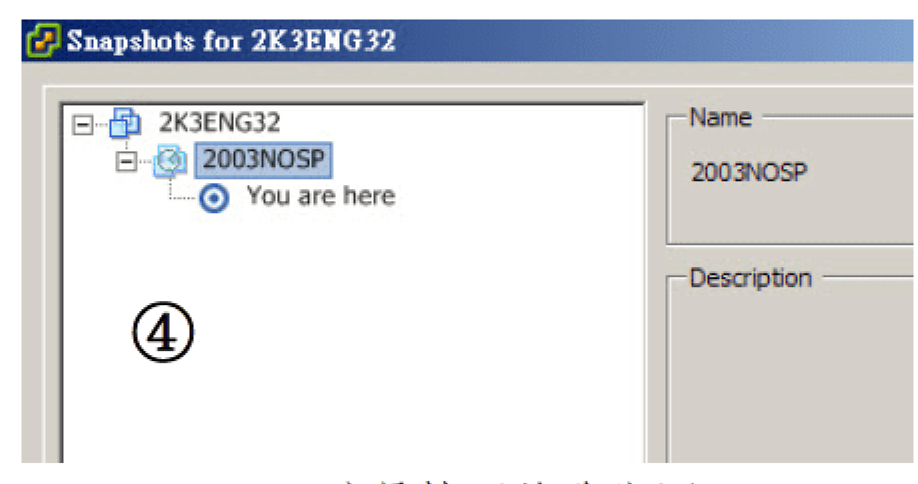
▲ 选择选项

3．此时弹出如下左图所示的画面，我们就键入 2003NOSP。此时再回到配置窗口时，我们就看到快照已创建。

4．我们选择 Inventory/Virtual Machine/Snapshot/Snapshot Manager 选项。 此时你会看到如下右图所示的画面。就是你当前的 Snapshot 路线图。我们可以在做好快照之后。开始安装 SP2，并且看看能否正确运行。



▲ 先创建一个快照



▲ 这是快照的路线图

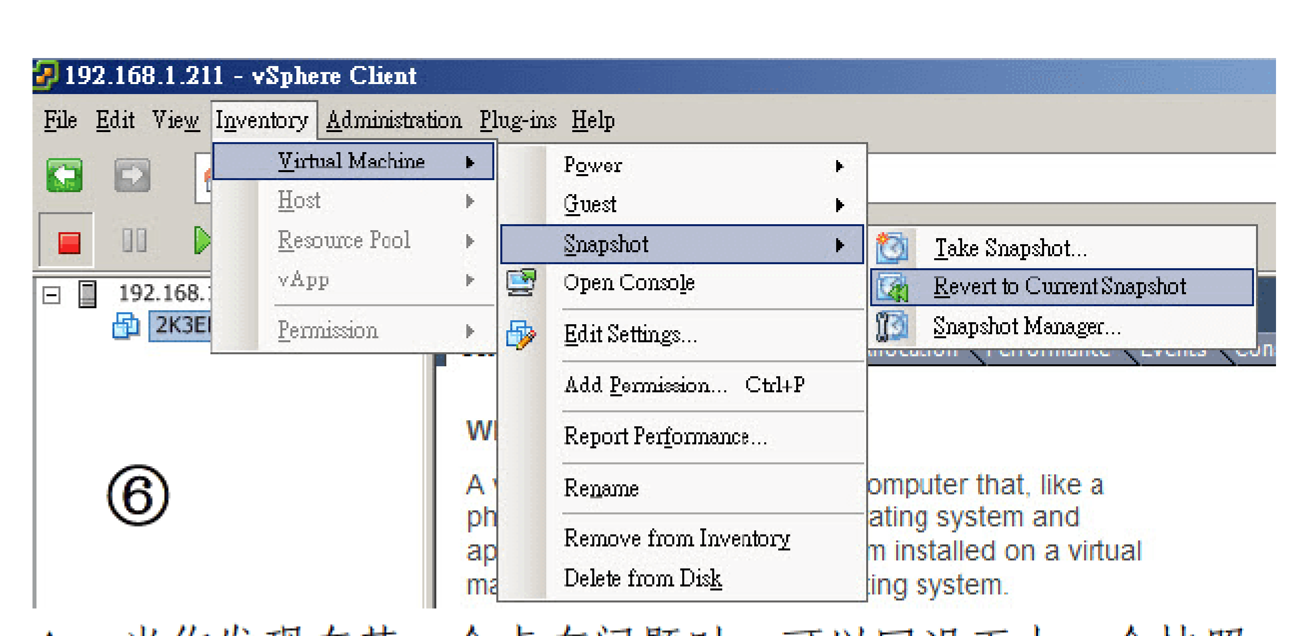
5．正确运行之后，我们再创建一个快照。称为 2003SP1。然后再开始进行测试。之后我们可以分别创建 2003SP2 的快照，来代表我们的进度。



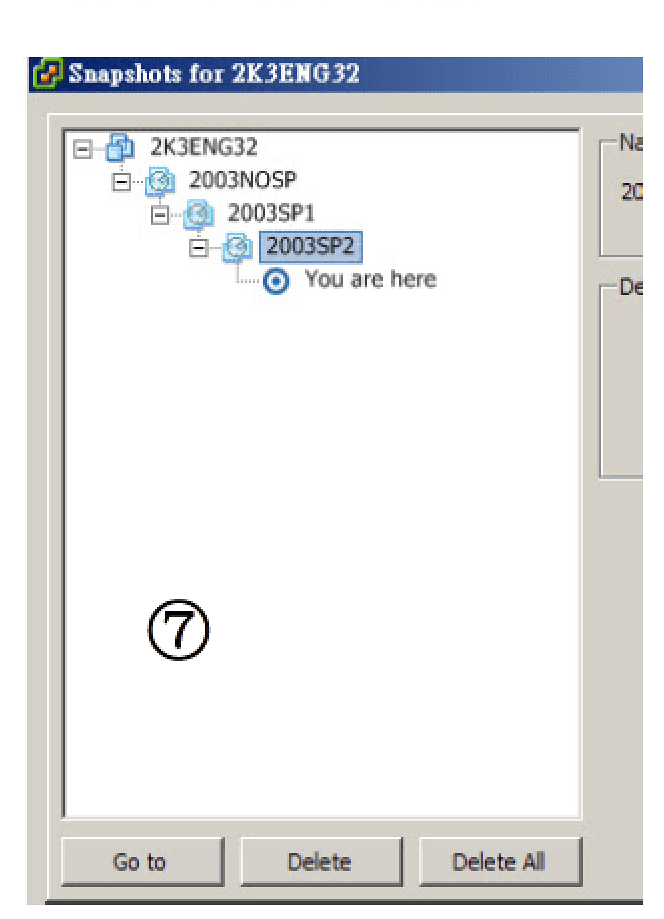
▲ 每一个点都可以配置一个快照

6. 当你在任何一点时，发现有问题，可以回退到上一个快照。手段只要选择 Inventory/ Virtual Machine/ Snapshot/ Revert to snapshot 选项。就会回退到上一个快照的创建点。

7. 当你发现每一个版本都正确无误之后，我们也可以一次将中间的快照都删除。



▲ 当你发现在某一个点有问题时，可以回退至上一个快照



▲ 可以将测试完的快照都删除

9.4.2 和 ESX 主机交换数据

大部分的计算机用户日常生活还是以 Windows 系列为主，但 ESX 是一个以 Linux 改装为主的 HypervCentersor，因此要和 ESX 的主机交换数据，必须理解 SSH 和 FTP 等命令，这对有大量的数据来说并不方便。此外在总控 ESX 主机上，虽然激活了 SSH，但我们也必须有一个较好的 SSH 客户端程序来管理 ESX 主机。我们在这一小段就来介绍两个最常用的工具 WinSCP 和 Putty。

1. WinSCP 的简介及使用

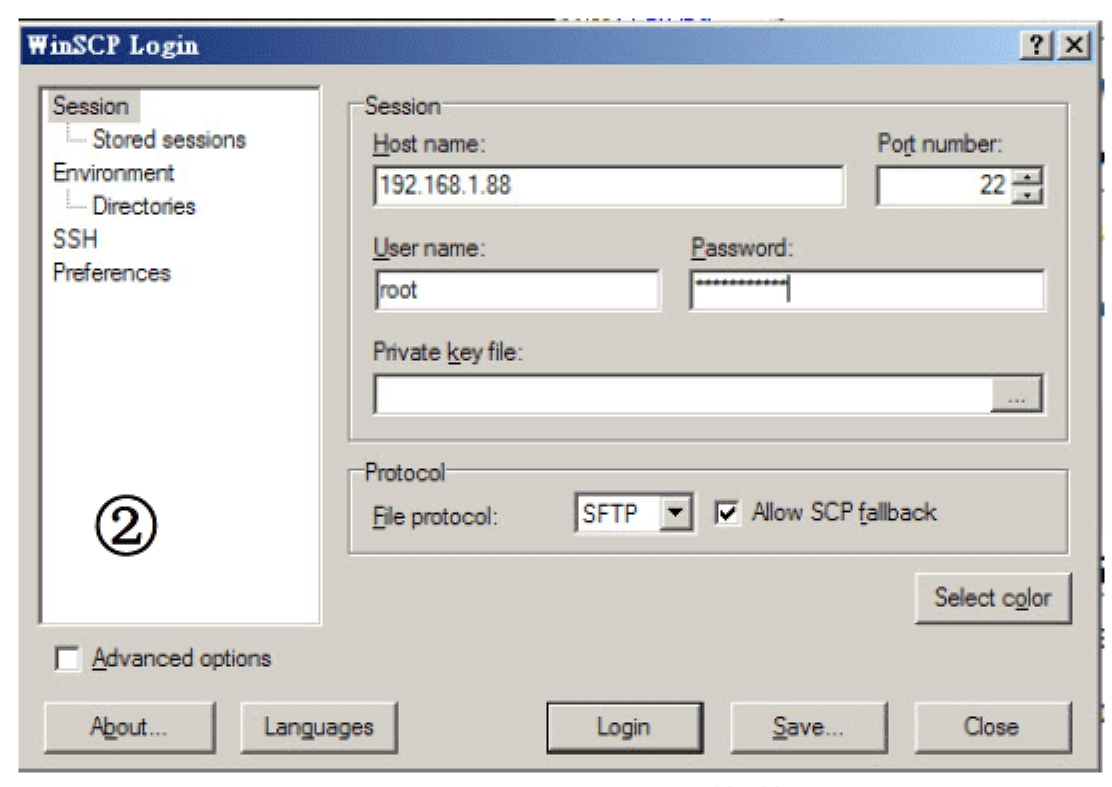
WinSCP 是一个窗口界面的 TFTP 程序，主要的用途就是以直白的拖拉方式和 Linux 主机之间交换数据。以 ESX/ESXi 来说，我们可以利用 WinSCP 和 ESX 之间提交 VMDK 或是 ISO 文件，在必要时也可以用来副本 ESX 主机中的数据。WinSCP 是一个自由软件，读者们可以到 http://winscp.net/eng/download.php 下载最新的版本，如果你下载的是免安装版，下载回来之后不需要安装可以直接使用。接下来就是使用 WinSCP 和 ESX 主机之间提交数据的例子。

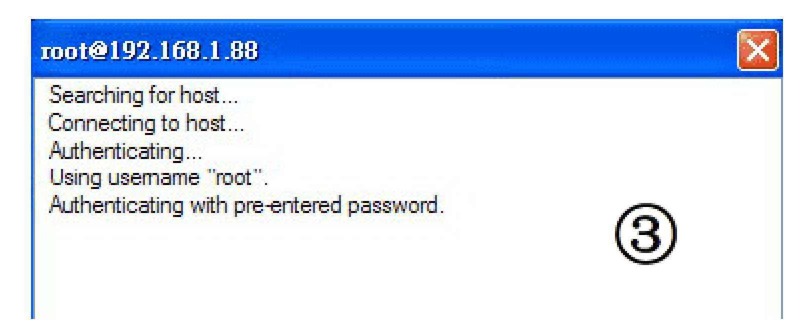


▲ 先到官方网站下载

▶ 使用 WinSCP 和 ESX 主机之间提交数据

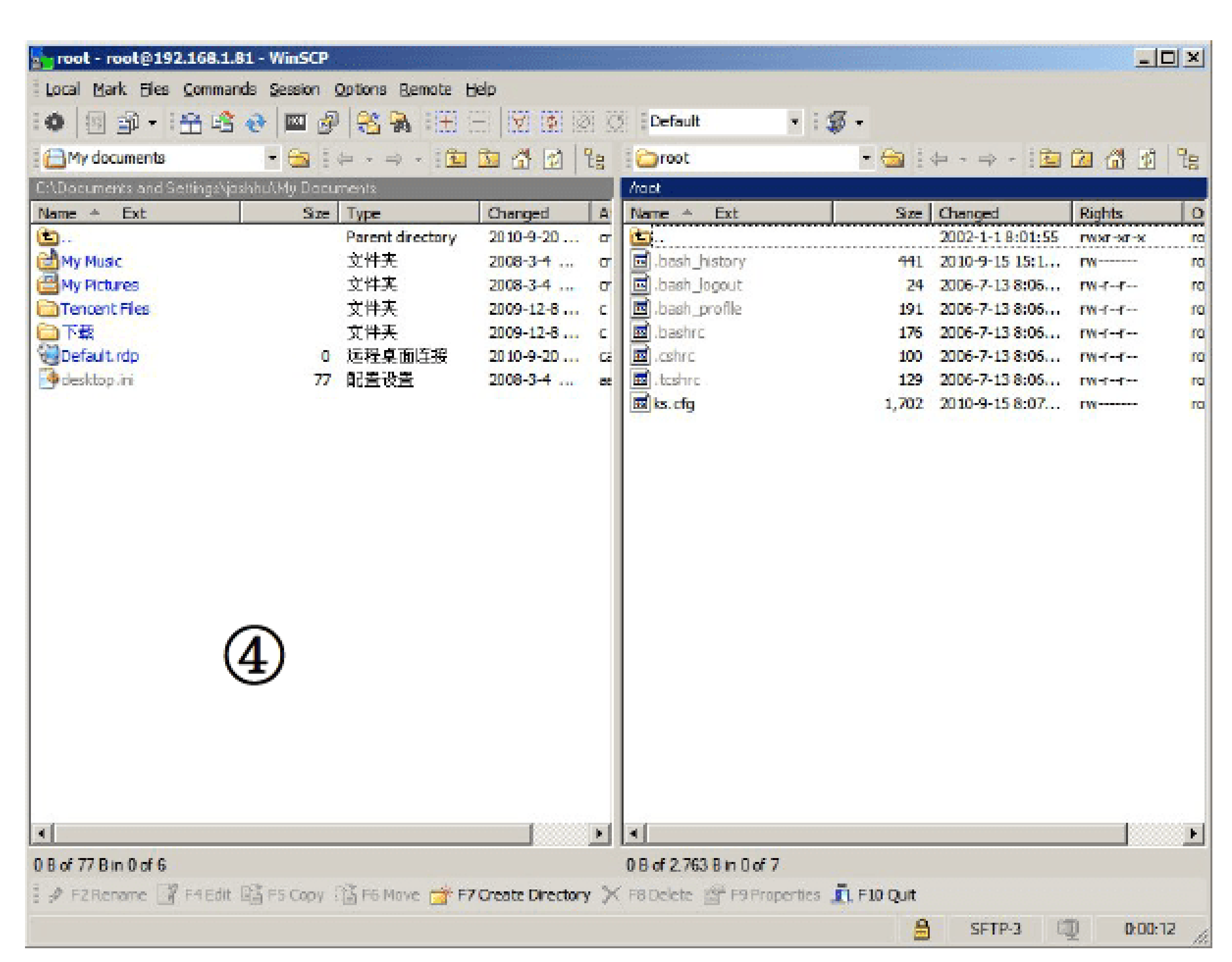
1. 首先确定你的 ESX 的 SSH 服务已经打开，请参阅本章第一小节的打开 ESX SSH 说明。
2. 进入 WinSCP，此时会弹出登录窗口。键入 ESX 主机的 IP 以及账号口令。在这个窗口中，只要键入 IP、账号、口令，其他一概不用动。键入完之后单击 Login 按钮登录。
3. 此时会进入登录窗口，登录 ESX 的主机。






▲ 键入 IP、账号、口令等等基本数据　　　▲ 此时会登录消息窗口

4. 当弹出如下图所示的画面时，意味着已经登录成功。你可以使用标准的 FTP 模式来传输数据，也可以多选、整个文件夹以及拖拉的方式来进行文件的克隆。



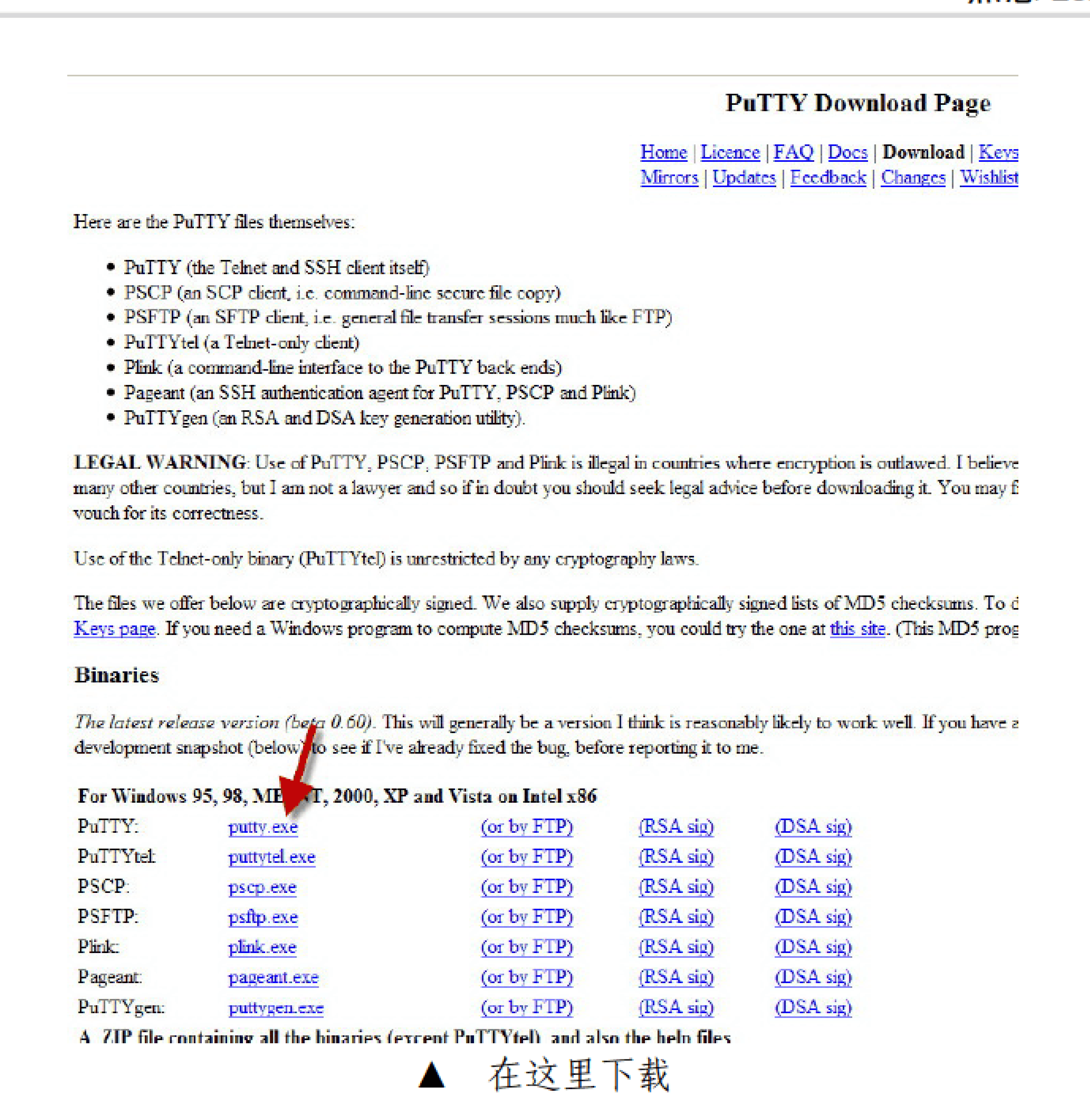
▲ 标准的 Windows 操作窗口

2. PuTTY 的使用

PuTTY 是标准的 Windows 下 SSH 程序。虽然我们已经有了 vSphere Client 来连入 ESX/ESXi 中，但使用点命令的方式是最快捷的方式。大部分的 Linux 客户端一定有 SSH 的客户端程序，但 Windows 下这一类的程序较少，而 PuTTY 则是最常见的。这一小段我们就来看看 PuTTY 这个 SSH 的使用。

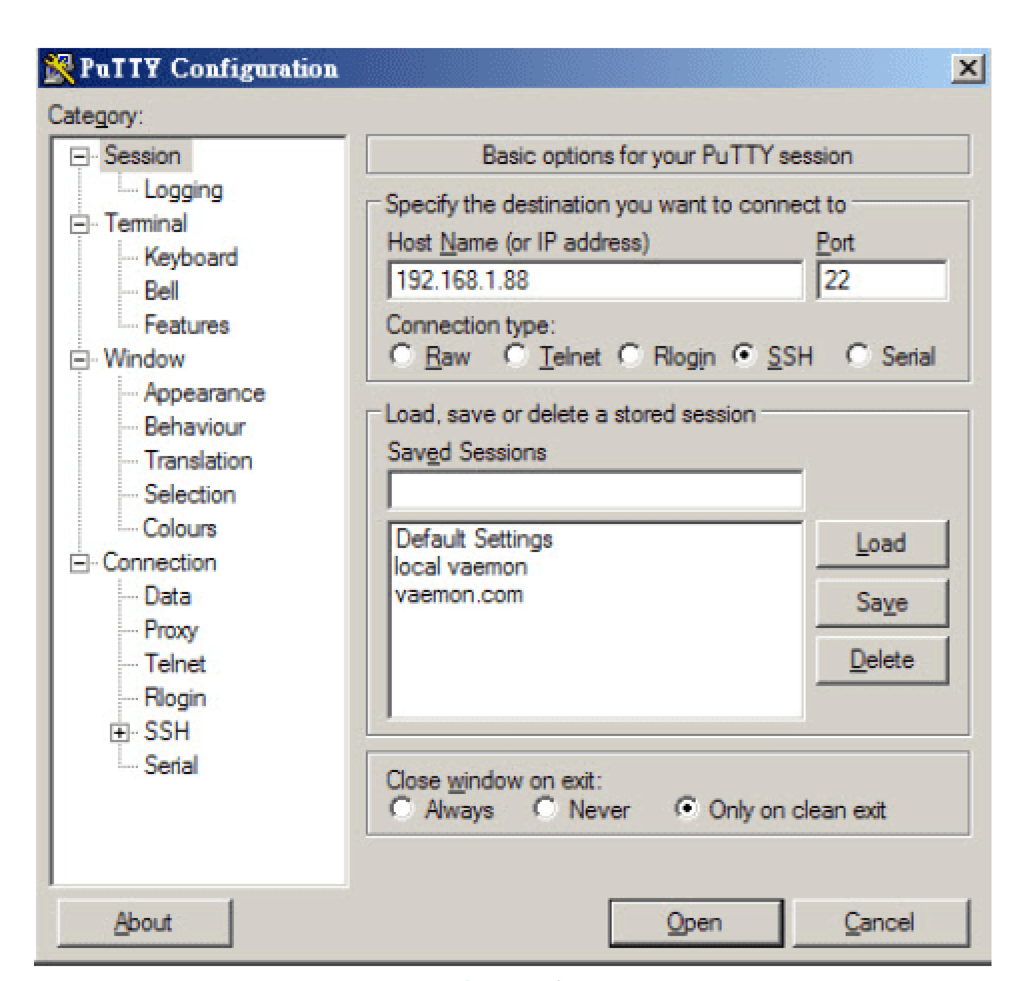
▶ 使用 PuTTY 连入 ESX

1. 我们可以到 www.chiark.greenend.org.uk/~sgtatham/putty/download.html 下载最新版的 PuTTY。下载回来是不用安装可以直接使用的。



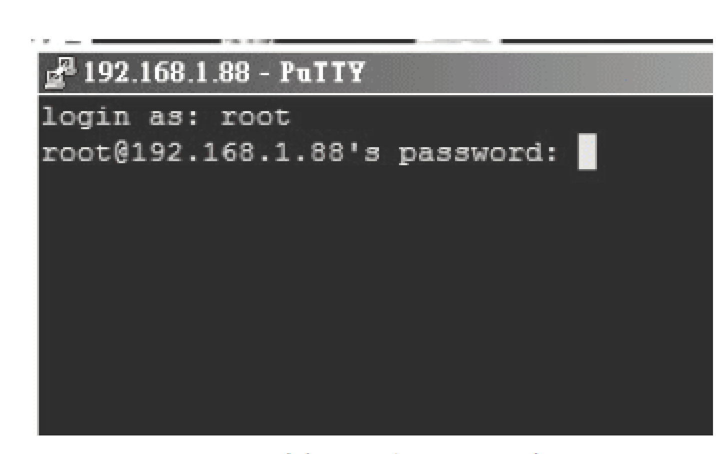
▲ 在这里下载

2．下载回来之后直接运行文件，会弹出如下图所示的画面。在该窗口中键入 ESX 的 IP 位置，并且单击 Open 按钮。



▲ 键入 IP 并且单击 Open 按钮

3．此时会弹出 ESX 的登录账号及口令部分。键入 root 以及 root 在 ESX 上的口令，就会登录到 ESX 主机中了。



▲ 键入账号口令

4．登录之后，你就可以使用 ESX 下的点命令进行操作了。

```
root@ESXonWK7:/
login as: root
root@192.168.1.88's password:
Last login: Wed Apr 21 13:17:50 2010 from mainwork
[root@ESXonWK7 ~]# ls
ks.cfg
[root@ESXonWK7 ~]# clear
[root@ESXonWK7 ~]# cd /
[root@ESXonWK7 /]# esxcfg-vswif
esxcfg-vswif <options> [<vswif>]
-a|--add                      Add vswif, requires IP parameters.
                              Automatically enables interface.
-d|--del                      Delete vswif.
-l|--list                     List configured vswifs.
-e|--enable                   Enable this vswif interface.
-s|--disable                  Disable this vswif interface.
-p|--portgroup                Set the portgroup name of the vswif.
-V|--dvs-name                 Set DVPort's DVSwitch name of the vswif.
                              Requires --dv-port-id to be specified
-P|--dv-port-id               Set DVPort's DVPort ID of the vswif.
                              Requires --dvs-name to be specified
-i|--ip <IP Spec>             Where <IP Spec> can be one of the following...
                              <X.X.X.X>      - to use static IPv4 address specif
ied
                              DHCP           - to enable IPv4 DHCP for address.
```

▲　可以在 SSH 使用 ESX 的专属命令

结 语

本章介绍了标准的 ESX 和 ESXi 的使用，并且将 ESX 和其他主机的连结也配置完毕。事实上，要将 ESX/ESXi 的功能完整介绍，整本书都不够说明，但从这一章的介绍，读者们也能在 ESX 上创建虚拟机了。整个 vSphere 架构的基础就是 ESX/ESXi 服务器，而 ESX 不管是从 vSphere Client 或是远程连入，甚至是主控台的操作，都是熟悉 ESX 的最佳途径，要理解整个虚拟机或 vSphere 的使用，本章是最重要的基础。

第 10 章 完全征服 vSphere 的网络原理

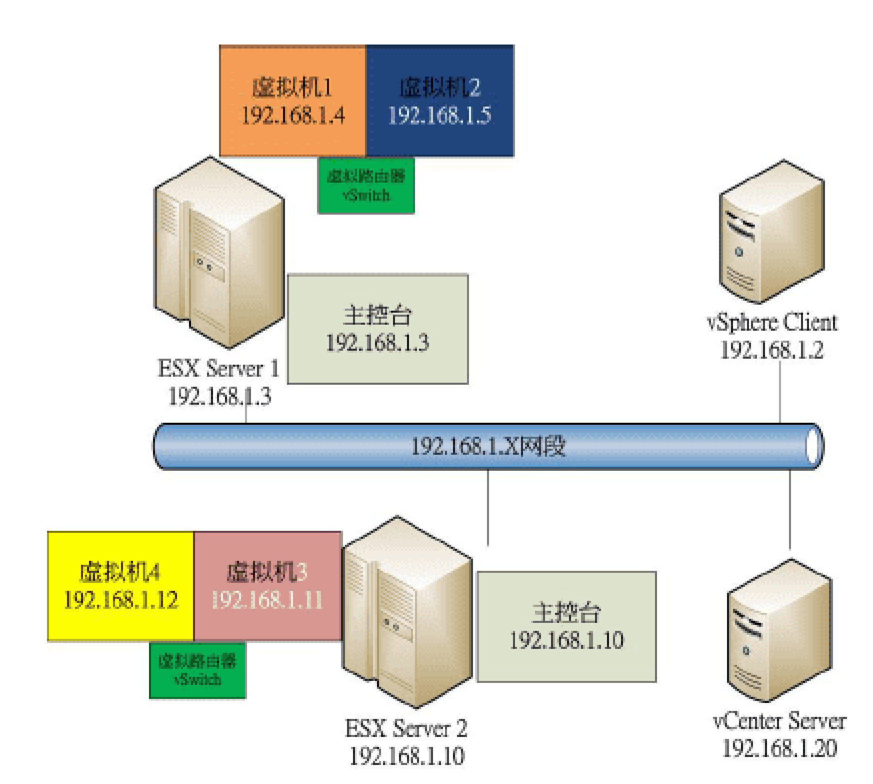
关键词：

- 理解单台，多台 ESX/ESXi 和单片，多片网卡之间的关系
- 理解 vSphere 的网络结构
- 理解 vSphere 上的网卡、交换机、通信组及功能之间的关系
- 理解主控台、VM 内核和虚拟机通信组是什么
- 理解多片网卡的使用方法
- 学会操作，配置，新建主控台，VM 内核和虚拟机通信组

ESX 作为原生架构的虚拟机产品，能在上面运行多个虚拟机，最需要关注的部件就是网络。没有网络的 ESX 上面有再多虚拟机也没有作用，但管理这些虚拟机之间的互相操作机制也同等重要。虚拟机被视为物理计算机时，多台虚拟机之间，以及虚拟机和 ESX 主机，甚至是网卡和其他的物理计算机或虚拟机之间的交互，也必须靠 ESX 上的机制来落实。这么重要的网络操作，ESX 是如何做到的？

10.1 理解 ESX 的网络原理

ESX 上的虚拟机被视为物理机，因此在网络的拓扑上，也需要使用最标准的网络概念。当前以 Ethernet 流行的网络组合方式，再加上互联网的 TCP/IP，出现了各个不同的网络连接方式和设备。物理网络设备如交换机和路由器，到了虚拟化的世界时，竟然也能用虚拟化的设备来代替，在这一小节中，我们就要来看看 ESX 虚拟机中的网络概念。



▲ 标准的 vSphere 网络架构，虚拟路由器占了极重要的角色

10.1.1 从物理网络到虚拟机网络

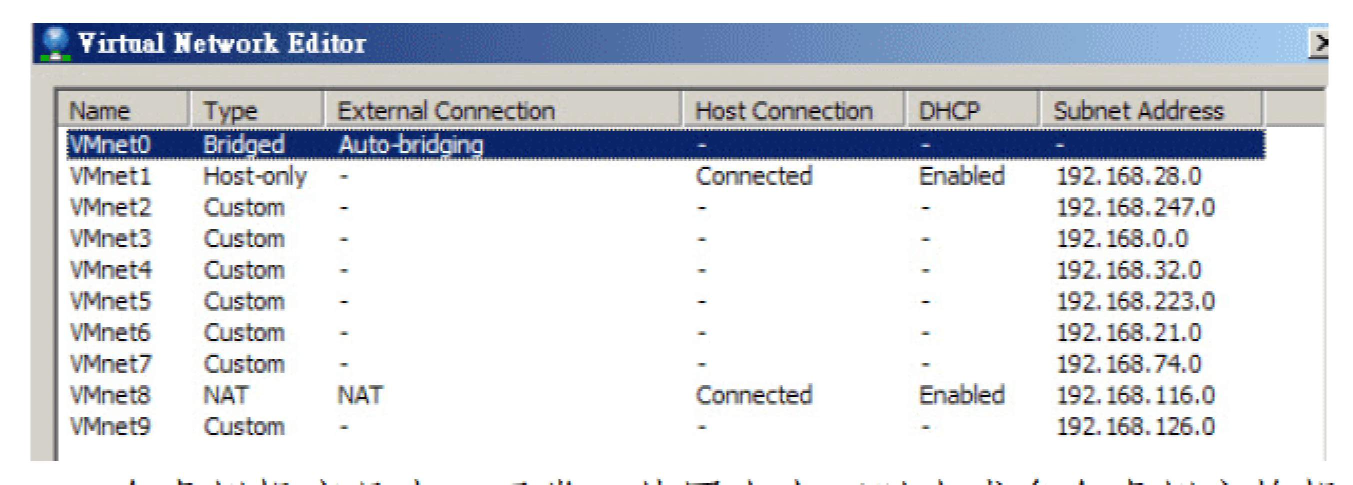
任何两台计算机之间的交流都必须靠网络，不管这两台计算机是物理机还是虚拟机。当邻近的两台以上的计算机组成了一个网络之后，我们称为一个局域网，局域网之间的计算机使用交换机互相联机交换数据。随着互联网的流行，不同局域网之间的计算机也必须进行交流，但由于地域性或是分工的制约，跨局域网之间的计算机不可能直接联机，因为跨局域网的计算机不可能联机到对方的交换机上，因此使用路由器或第三层交换机，将同一个局域网计算机上的 IP 或 MAC 用路由设备的 IP/MAC 来取代，并且进行拆解包的动作来交换数据，这就是基本的网络及互联网的概念。



▲ 真实网络环境使用的设备也是真实的，如交换机

1. ESX 上虚拟机的局域网

在 ESX 上，由于多台虚拟机和多个网卡配合，在同一个网段的虚拟机中，组成了局域网（通常是以一个 IP 段来划分），此时就需要一个“交换机”来让这些不同的虚拟机之间进行联络。ESX 提供了虚拟机的交换机配置，交换机不但可以连接不同的虚拟机，更可以提供物理机上的主控台和存储设备等数据。

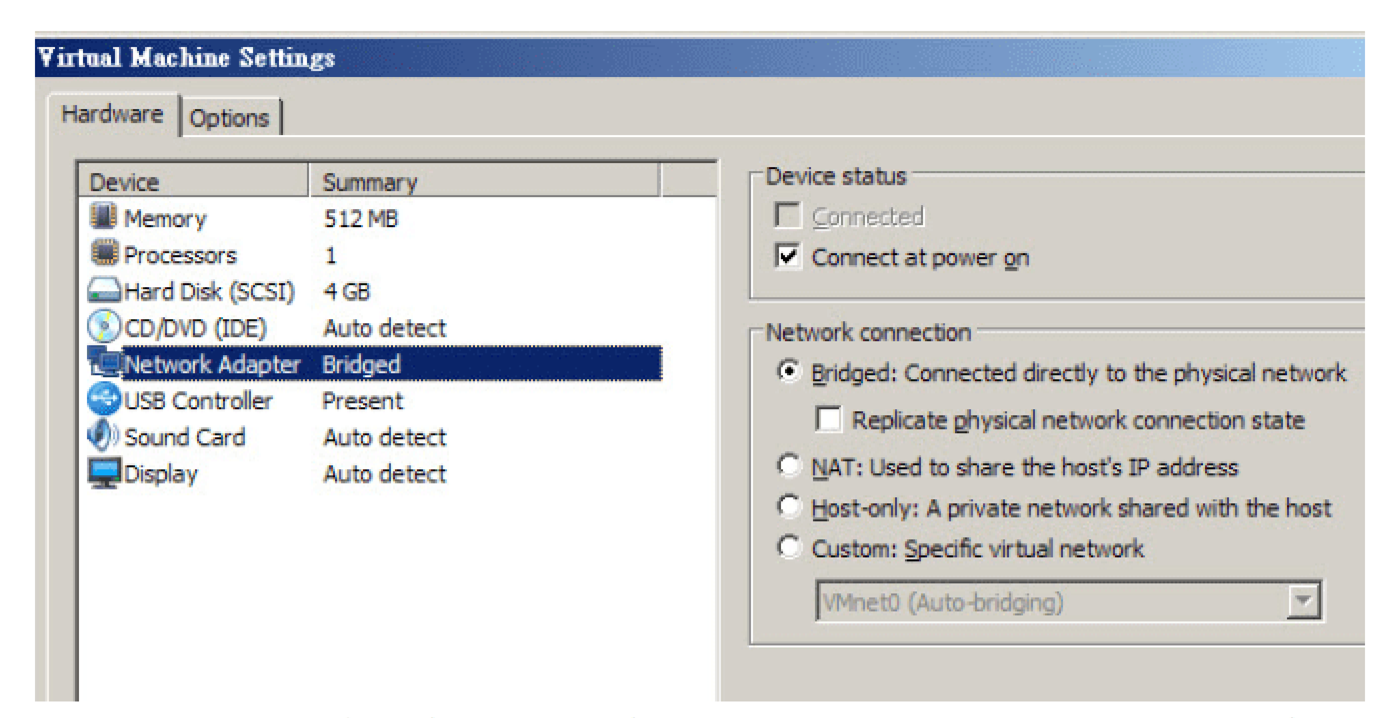


Name	Type	External Connection	Host Connection	DHCP	Subnet Address
VMnet0	Bridged	Auto-bridging	-	-	-
VMnet1	Host-only	-	Connected	Enabled	192.168.28.0
VMnet2	Custom	-	-	-	192.168.247.0
VMnet3	Custom	-	-	-	192.168.0.0
VMnet4	Custom	-	-	-	192.168.32.0
VMnet5	Custom	-	-	-	192.168.223.0
VMnet6	Custom	-	-	-	192.168.21.0
VMnet7	Custom	-	-	-	192.168.74.0
VMnet8	NAT	NAT	Connected	Enabled	192.168.116.0
VMnet9	Custom	-	-	-	192.168.126.0

▲ 在虚拟机产品中，通常一片网卡上可以生成多个虚拟交换机

2. ESX 虚拟机上的跨网络

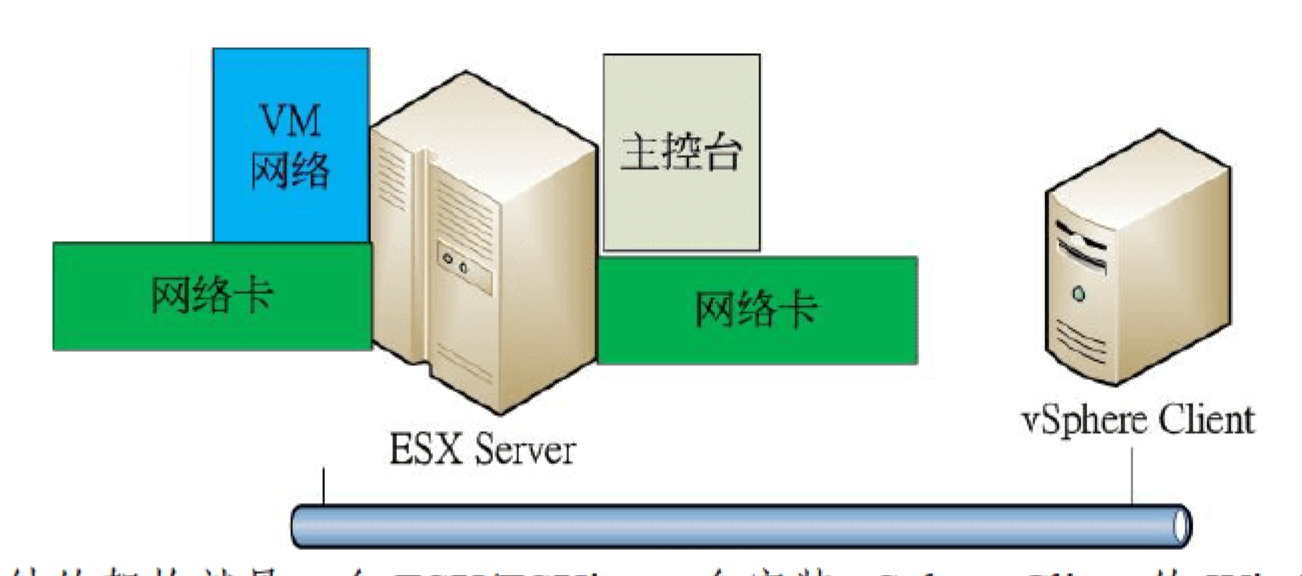
当同一台物理机上拥有第二片网卡，而这片网卡又连上另一个网段时，捆绑在这两片网卡上的不同虚拟机之间必须交换数据，此时需要一个“路由器”或“三层交换机”。此外当两台物理机上的虚拟机需要交换数据时，也需要一个物理交换机和虚拟路由器的配合，有时还需要给定 DHCP 等服务，因此 ESX 也提供了一个设置路由器的功能。



▲ 较高级的虚拟机产品本身还可以提供 DHCP 或 NAT 的功能

10.1.2 理解 vSphere 上的网络结构

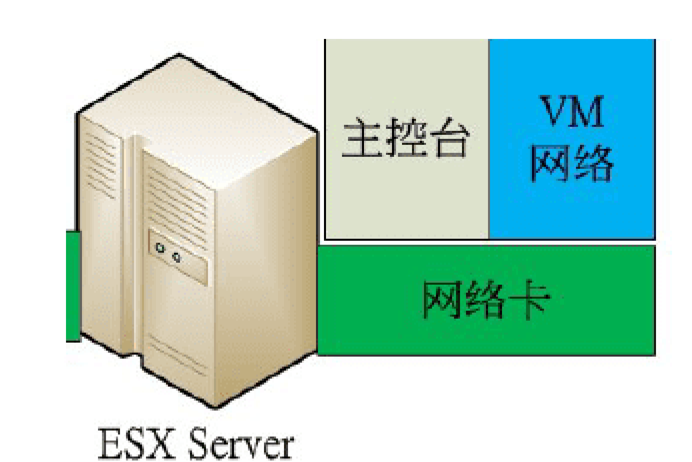
不管是单台的 ESX/ESXi 或是完整的 vSphere 架构，ESX/ESXi 服务器的配置从最少一片网卡到多片网卡，都需要和物理网络进行沟通，才能使用 vSphere Client 进行总控。因此最单纯的 vSphere 网络架构，就是一个物理网络中的两台计算机（一台 ESX/ESXi 及一台 vSphere Client）。但是在加入了存储设备或是其他的网络部件后，整个网络架构也变得复杂了。在 ESX/ESXi 上，网络功能分为主控台、内核、VM 三大类，我们先介绍一下这些拓扑，在本章稍后会有更详细的介绍。



▲ 最单纯的架构就是一台 ESX/ESXi，一台安装 vSphere Client 的 Windows 系统

1. 单台 ESX/ESXi 的网络架构

单台 ESX/ESXi 还是需要一台 vSphere Client，此时 ESX/ESXi 至少要有一片网卡，并且提供总控台一个 IP，这个 IP 要和 vSphere Client 位于同一个 IP 段才能总控。此外这台 ESX/ESXi 上的网卡也要负责提供 VM 联机至同一个 IP 网络段，因此是一片网卡并发负责主控台和每一个 VM 的路由功能。

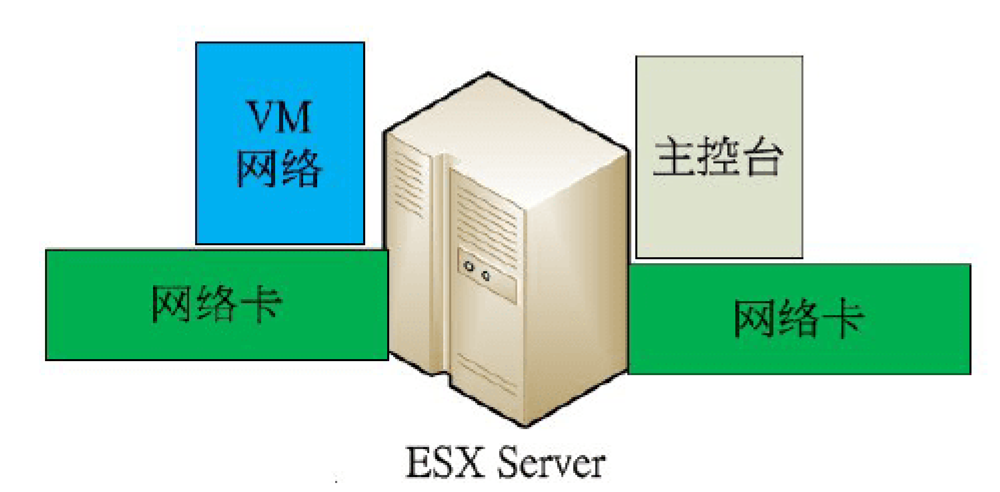


▲ 单台 ESX/ESXi 的单片网卡提供了主控台连接及 VM 的网络功能

2. 单台 ESX/ESXi 多片网卡架构：功能分离

这和前面的情况一样，但这时如果单台

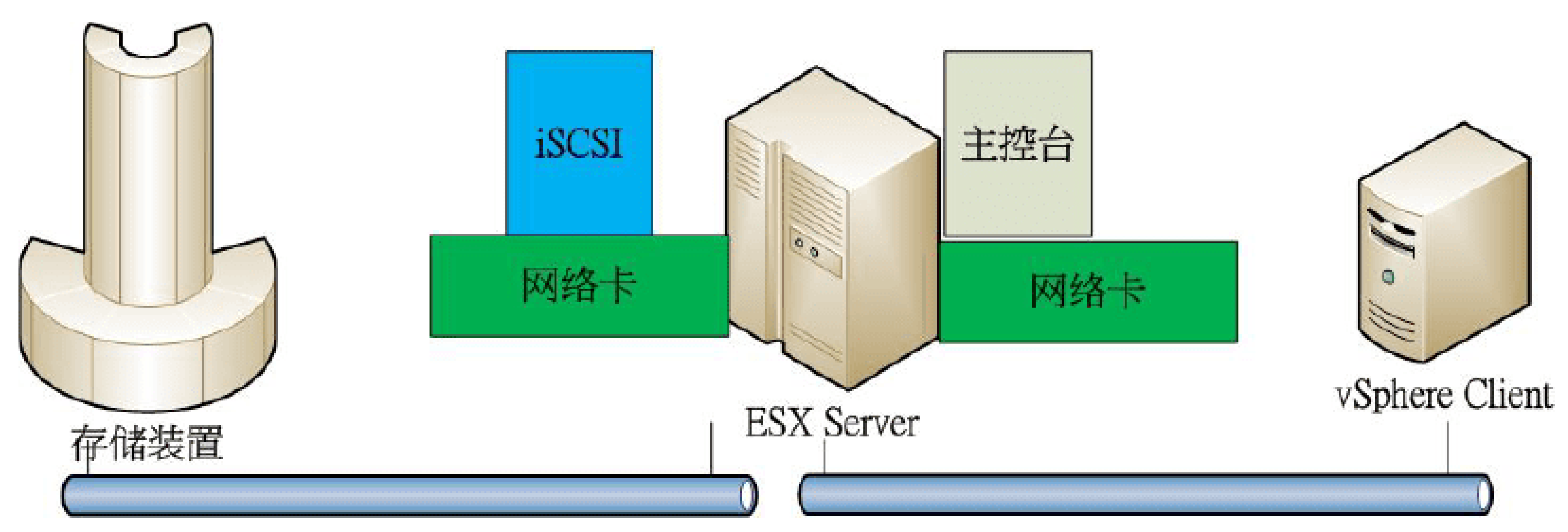
ESX/ESXi 上有多片网卡，能做的组合就很多了。首先我们可以将一片网卡负责作为和 vSphere Client 总控用，另一片网卡作为 VM 的路由功能。这么做的好处就是能让网络性能不会受到某一片网卡的影响。



▲ 如果有两片网卡，可以将主控台和 VM 功能分开，VM 的性能较不受影响

3. 单台 ESX/ESXi 多片网卡架构：VLAN

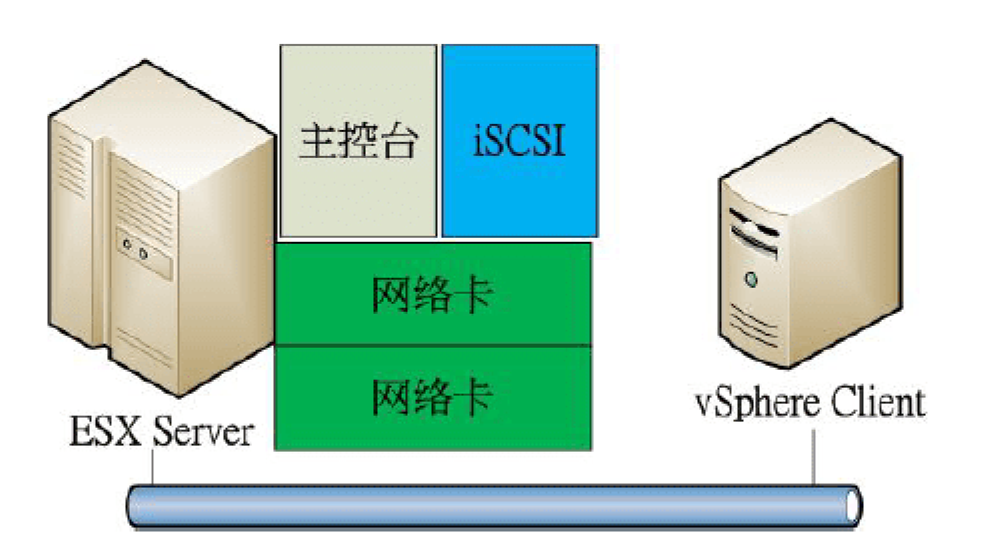
如果你的网络路由器有 VLAN 的功能，就可以让单台 ESX/ESXi 上不同的网卡连接到不同的 IP 段。通常这么做的场合是有存储设备连上不同的 IP 段时，为了区分存储设备和真正使用的网络，会将网络相关的通信（如主控台或 VM 的路由功能）放上一个 VLAN，而将存储功能（如 iSCSI 或是 NFS）放上另一个 VLAN，因此会用到两张网卡。



▲ 我们也可以将多片网卡设成不同的网段来连接到不同的应用上

4. 单台 ESX/ESXi 多片网卡架构：网卡合并

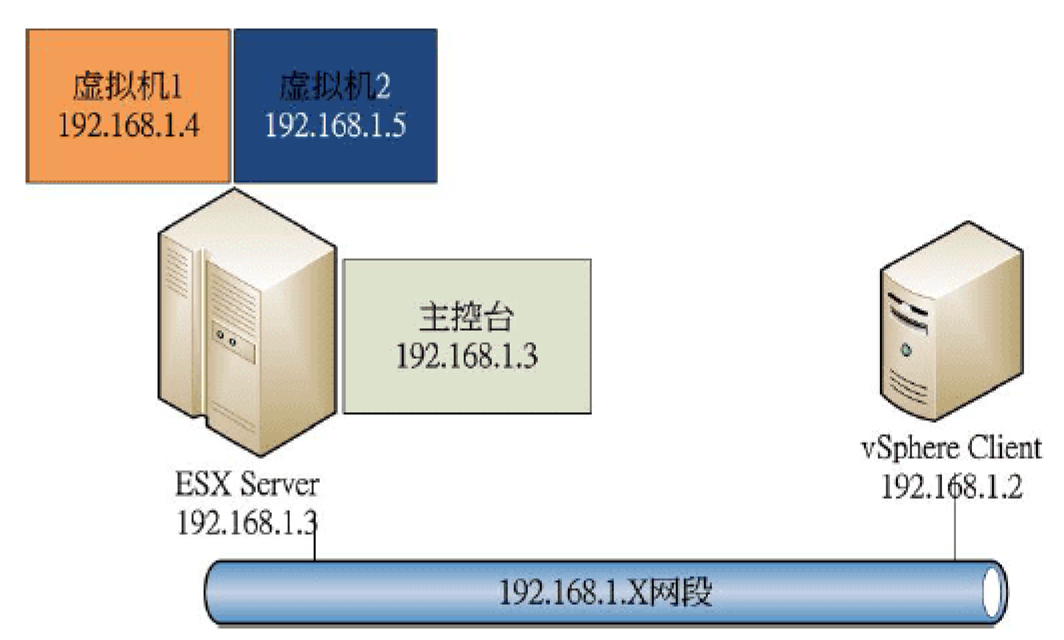
在需要加快速度或着重安全的场合，多片网卡可以合并成一个网络界面来给单台 ESX/ESXi 使用，称为 NIC Teaming。事实上如果你的服务器有四个以上的网卡，在安装 ESX/ESXi 时，系统会自动将其中错开的两片网卡组合成一个 NIC Teaming 来给主控台使用。这样的好处就是可以在某一片网卡坏掉时，系统能继续常规任务。



▲ 多片网卡做成一个 vSwitch 可以确保安全上无虑

5. 多台 ESX/ESXi 多片网卡架构

在有了多台 ESX/ESXi 之后，物理服务器之间会有许多功能要靠网卡落实，因此必须要有多片网卡，才能彻底发挥 VMotion、HA 或 FT 功能，我们在稍后的章节会有完整的示例。



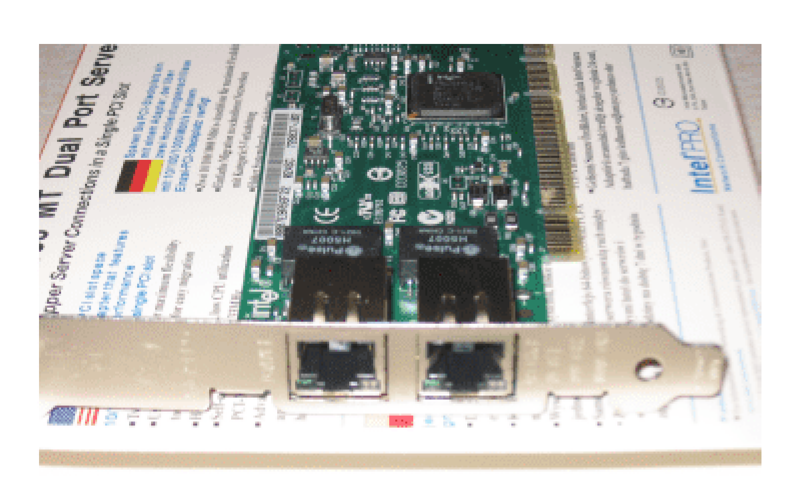
▲ 多台 ESX/ESXi 的多片网卡才能落实 vSphere 中的高级功能

10.1.3 ESX 上的网络部件

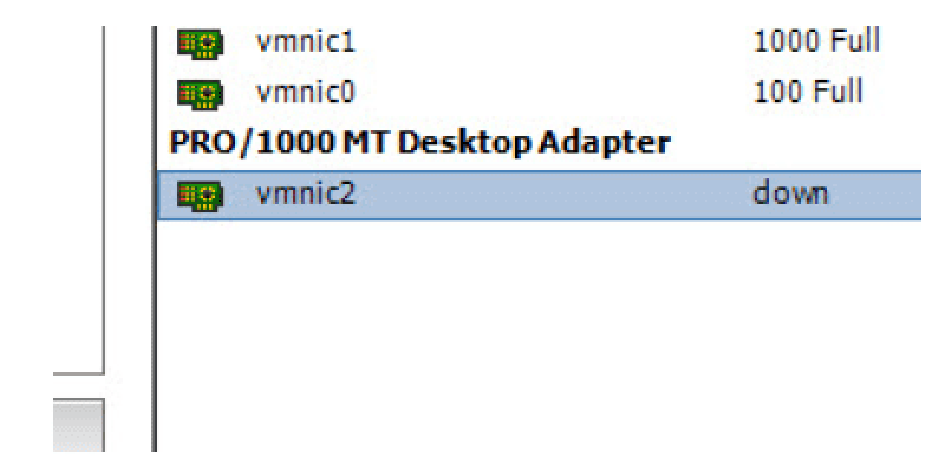
为了要达成上述的功能，又必须兼顾 ESX 物理机的操作和管理，因此 ESX 在网络上的规划比 Hyper-V 或 VMware Workstation 要复杂很多。虽然功能复杂，但观念和使用上却很简单，我们在这一小节就来看看 ESX 网络中最重要的部件的说明。

1. 最基本的设备：物理网卡——vmnic

在 ESX 中，你可以安装非常多物理网卡，视你机器的物理制约而定。在 ESX 中的物理网卡是所有网络设备的根本。物理网卡可以是接上物理网络的交换机或路由器，也可以是 ESX 服务器群集之间的心跳线，或是网络存储 iSCSI 的设备。在 ESX 中，物理网卡的设备名称为 vmnic。第一片物理网卡为 vmnic0，第二片则为 vmnic1，以此类推。我们在安装好 ESX 时，其实第一片 vmnic0 也会安装完毕。



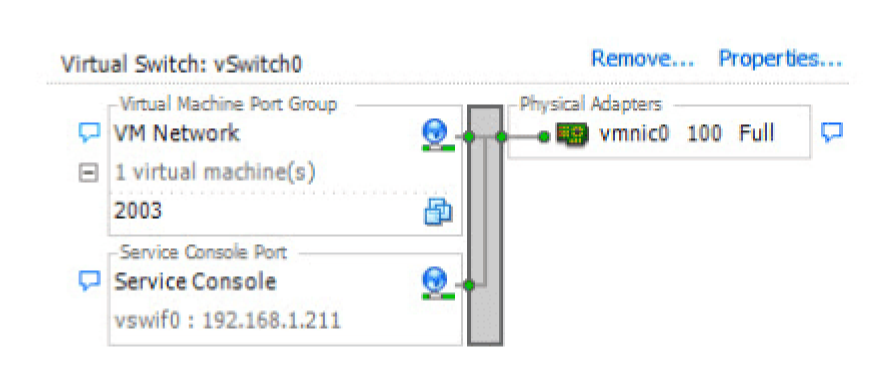
▲ 一个网络口就称为一个 vmnic，一片网卡有两个网口就有两个 vmnic



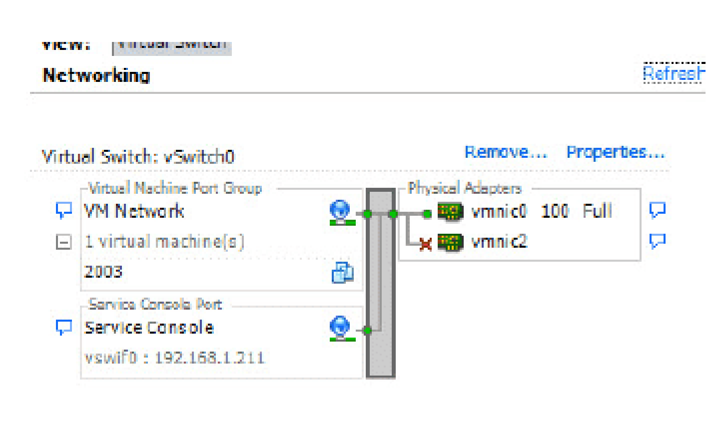
▲ 物理网卡在 ESX 上称为 vmnic

2. 虚拟交换机：vSwitch

虚拟交换机（Virtual Switch）是由 ESX 的内核提供的一个虚拟交换机。这个交换机主要的目的就是让不同的虚拟机和管理界面之间互相连接。在安装好 ESX 之后，系统默认会创建一个新的虚拟交换机 vSwitch0，这个交换机的功能主要是提供主控台、虚拟机联机等。要注意的是，一个虚拟交换机可以由一片或多片 vmnic 所组成。



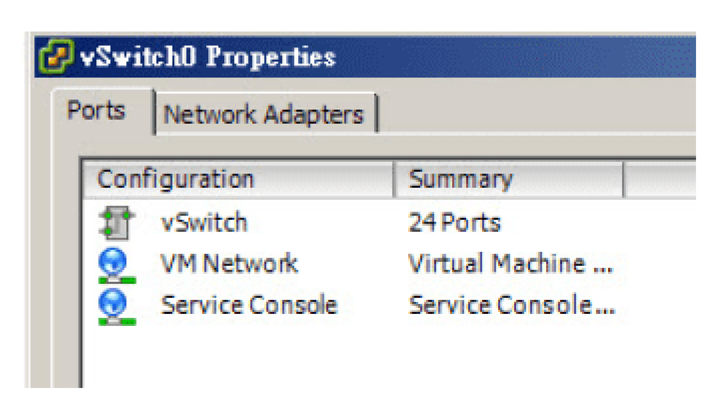
▲ 在 ESX 中，虚拟交换机是最重要的 Container，其承载了所有功能



▲ 可以由多片网卡组合而成

3. 通信端口/通信端口组：Port/Port Group

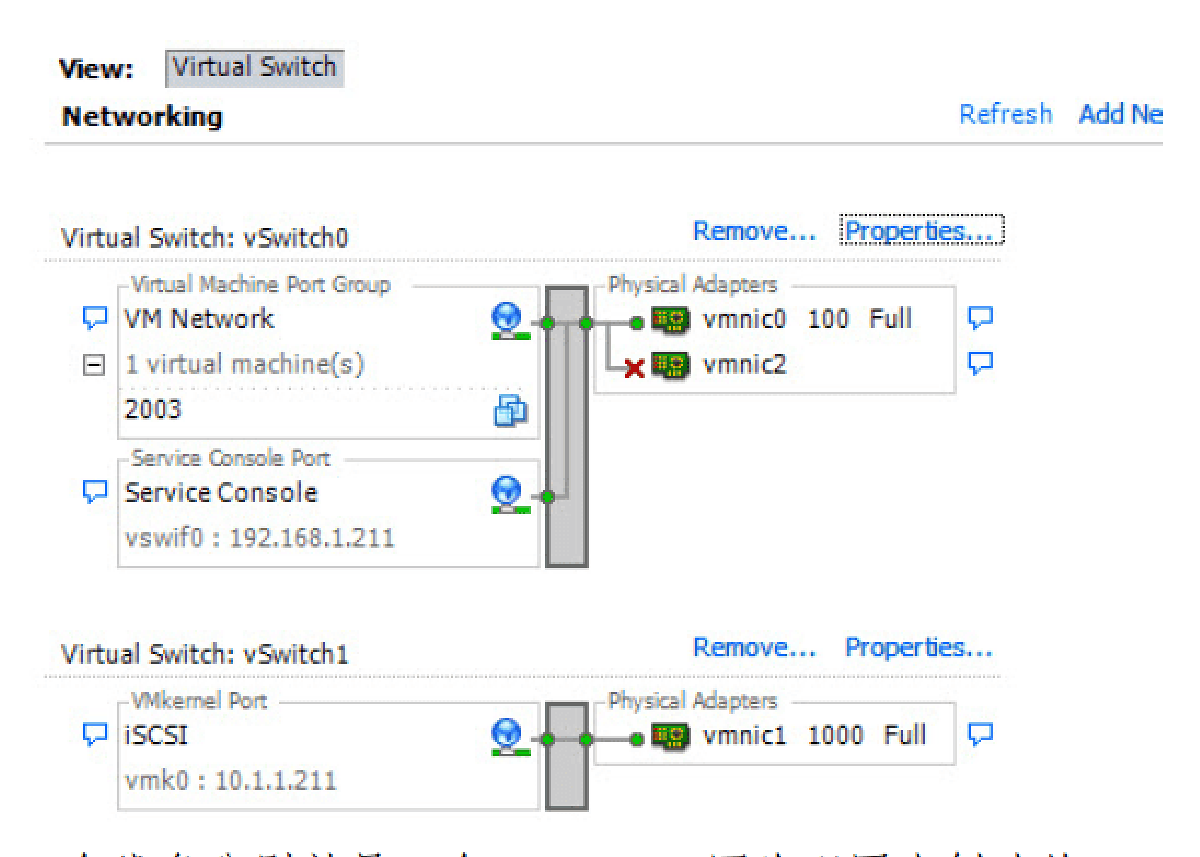
在一个虚拟交换机中，可以拥有不同的功能，这些功能都要靠通信端口和通信端口组来落实。通信端口/通信端口组是虚拟交换机中的逻辑对象，主要的目的就是针对不同的功能提供服务。在 ESX 中，通信端口/通信端口组提供了最重要的三个功能，分别是主控台（Service Console）、ESX 内核（VMKernel）和虚拟机通信端口组（Virtual Machine），所有 ESX 的网络操作，就是集中在这些通信端口/通信端口组和虚拟交换机、物理网卡之上。



▲ 一个 vSwitch 上的所有功能是由 Port/Port Group 进行通信

4. vSwitch 和物理网卡的关系

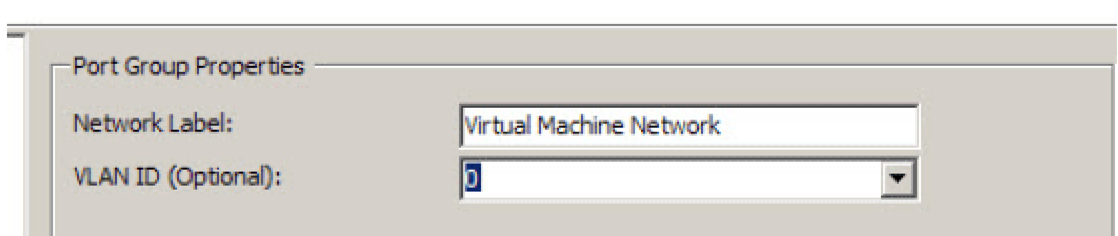
在 ESX 中，每一个网络的单位划分都是以 vSwitch 为主。我们在 vSphere Client 中可以看到一个深色的方帧就意味着一个 vSwitch。为了具体需要，vSwitch 和物理网卡 vmnic 之间会有不同的排列组合，进而提供更多的 ESX 服务，如集群、备援等。vSwitch 和物理网卡可以拥有一对一、多对一的关系。举例来说，一个 vmic 上可以有一个 vSwitch，而一个 vSwitch 可以是由一个 vmic 组成，也可以是由多个 vmnic 组成，但前提是不同的交换机无法共享同一个网卡。



▲ 一个线条分别就是一个 vSwitch，图为双网卡创建的 vSwitch

5. VLAN 的部件

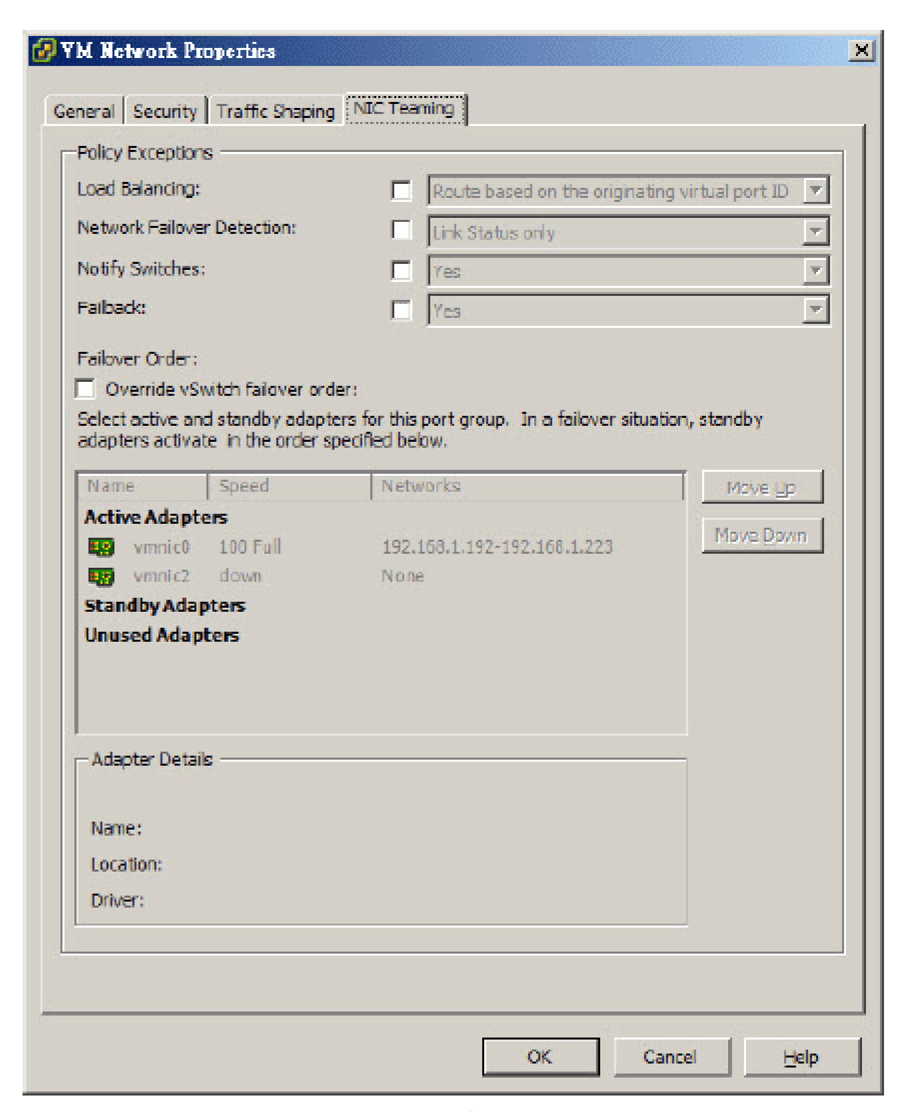
在 ESX 中尚有其他的部件，但主要都是和 VLAN 有关。VLAN 的观念和虚拟机或物理机无关，主要是划分网络以及安全管理上有关的服务。但在 ESX 中，也因为路由器以及交换机的虚拟化，使得 ESX 也能在虚拟机的基础上再将网络的分段抽象化，要让 vSphere 能支持 VLAN，需要物理交换机的支持才行。



▲ ESX 的 VLAN 功能处处可见

6. NIC Team

NIC Team 的功能就是将不同的网卡集成至同一个逻辑信道上。在 vSphere 中，NIC Team 的功能变多了，除了可以让速度增快之外，也可以用来做 Failover，我们在本书稍后的高级功能中会有提到。



▲ 在 vSphere 上，NIC Teaming 的功能非常完整，可以配置备援或提速

10.1.4 在 VM 上的网卡

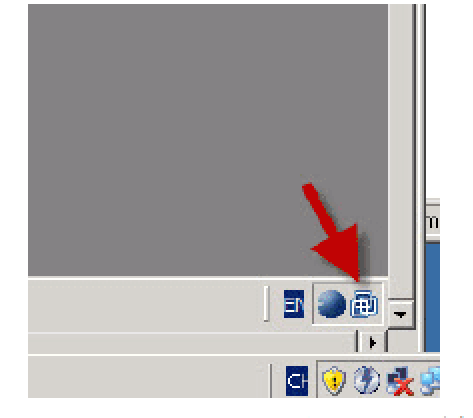
在 10.1.3 小节中，我们提到了 ESX/ESXi 上的部件。事实上在安装完 VM 之后，VM 也必须和网络沟通，因此每一个 VM 上都会有一个虚拟的网卡。视其功能而定，VM 上的网卡也有下面几种：

1. vlance 网卡

在安装 VMware Tools 之前的虚拟机客户端操作系统上使用的网卡称为 vlance。这片网卡只能提供单纯的网络访问，无法配置更多的功能，但是在没有提供 VMware Tools 的客户端操作系统中，这也是唯一的网卡选项。

2. vmxnet 网卡

当你在任何虚拟机操作系统中安装了 VMware Tools 之后，虚拟机中的网卡就会变成 vmxnet 这片网卡。vmxnet 是一个高性能的虚拟网卡，不但速度快，上面能配置的参数也很多，因此不管是什么虚拟机的操作系统，在 ESX 上完装完毕之后，一定要安装 VMware Tools。



▲ 在安装完 VMware Tools 之后，就会弹出 vmxnet

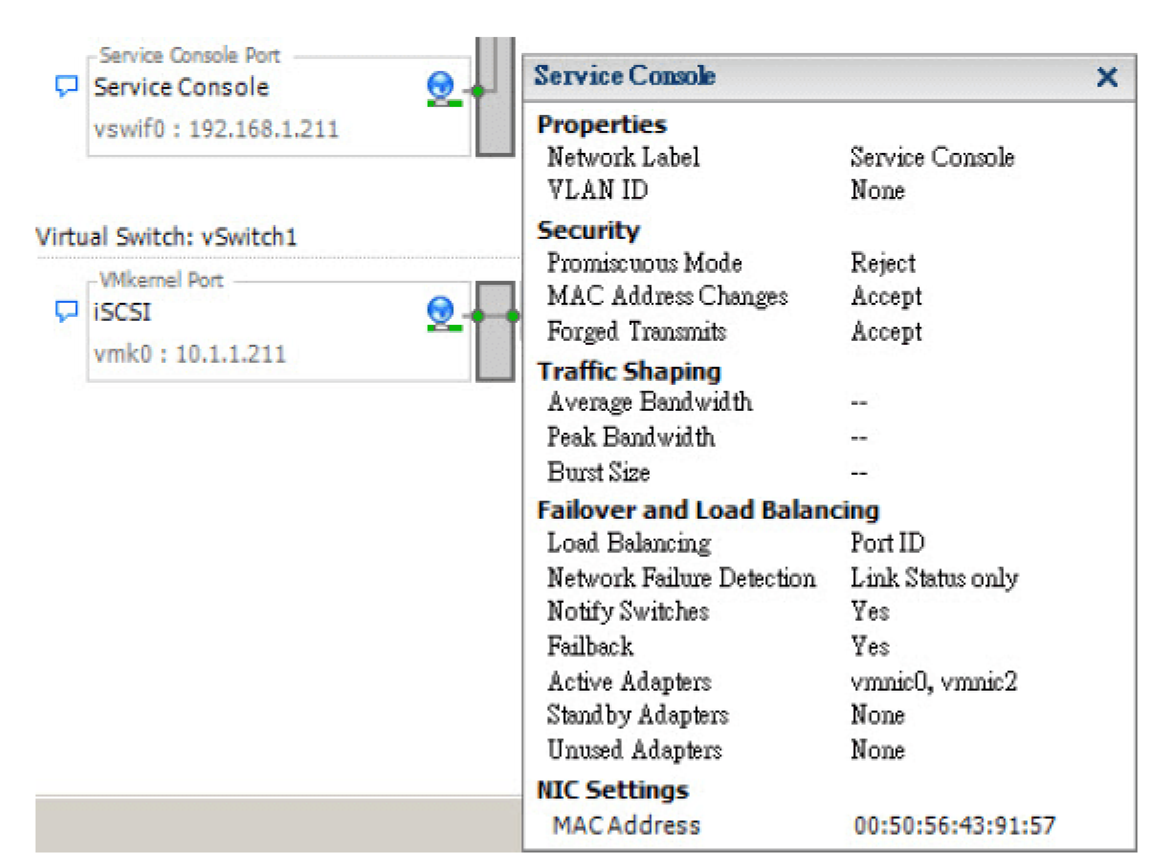
10.1.5 通信端口/通信端口组的详细说明

在 ESX 中，每一个 vSwitch 都可以包含不同的通信端口/通信端口组，这些通信端口和通信

端口组一共分成三类，下面就是详细介绍。

1. 主控台通信端口（Service Console）

主控台是一个很特殊的通信端口/通信端口组。主要的目的就是让系统管理员能使用 vSphere Client 连入管理。这个通信端口/通信端口组必须拥有一个 IP。在 ESX 中，主控台通信端口/通信端口组通称为 vswif，第一个创建的主控台称为 vswif0，以后以此类推。



▲ 这是 ESX 的主控台

2. ESX 内核（VMKernel）通信端口

ESX 内核（VMKernel）也是一个特殊的通信端口/通信端口组，其必须拥有一个固化的 IP，主要的目的是让多台 ESX 物理机或网络存储设备能连入，如 iSCSI、NAS 或是 NFS 等。我们在稍后提供的 VMotion、HA 等功能都必须通过这个通信端口来进行，因此这个通信端口通常也是和外网连接的。在 ESX 中，VMKernel 通信端口称为 vmk，第一个创建的 VMKernel 称为 vmk0，以后也以此类推。在新版的 vSphere 中，VMKernel 的功能已经越来越多了，下面就是几个最常见的。

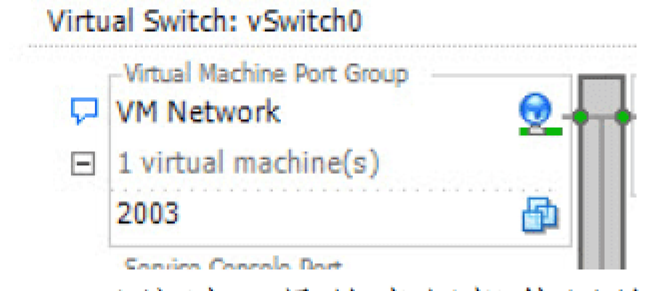
▶ **常见的 VMKernel 功能**

- 虚拟机在物理机之间移动的 VMotion。
- 访问网络存储设备的 iSCSI 或 NFS。
- Fault Tolerance 中的 LOG 表项。

▲ VMkernel 主要是给 iSCSI 和 vSphere 的高级功能使用

3. 虚拟机通信端口组（Virtual Machine）

虚拟机通信端口的用处就是让 ESX 上的虚拟机能连上另一个交换机，这另一个交换机很可能也是另一个虚拟交换机，但也可能是一个物理交换机。举例来说，我们可以创建一个虚拟机通信端口组让某一台 ESX 上的所有虚拟机之间形成一个封闭的局域网，但我们也可以创建另一个虚拟机通信端口组让 ESX 上的虚拟机连上一个物理交换机，让虚拟机和物理机之间能互通，因此虚拟机通信端口组主要的对象就是 ESX 上的虚拟机。

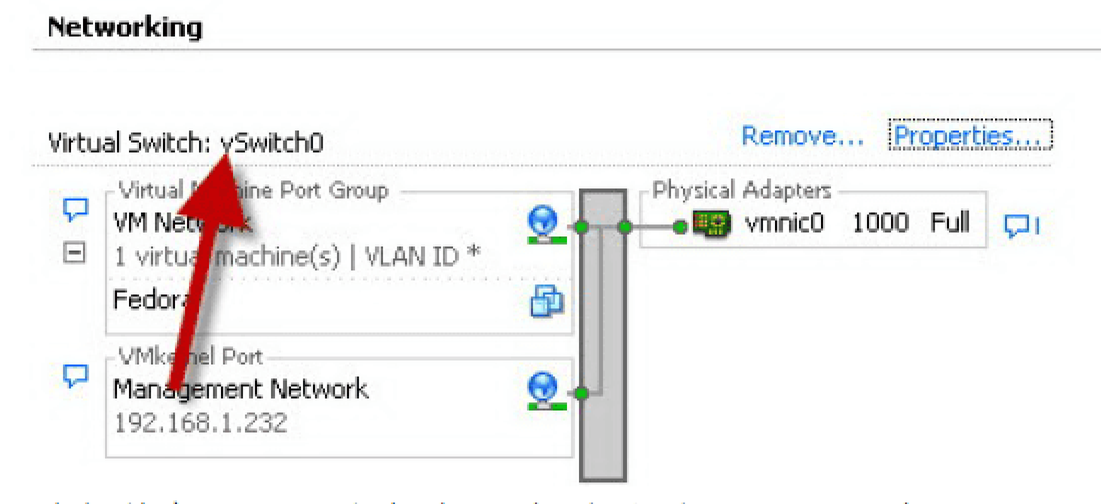


▲ 通信端口是给虚拟机使用的

> **注意**
>
> ESX 和 ESXi 的不同
>
> 由于 VMware ESXi 严格来说并没有主控台，必须使用 vSphere Client 或简单的总控项来操作，因此在其网络配置中，并无法加入主控台通信端口。在系统创建时，会有一个 Management Network。在新加入 Networking 时，也只能选择虚拟机的通信端口组或是 VMkernel，其他地方则和 ESX 完全一样。
>
> ▲ ESXi 使用的则是管理账户

10.2 创建及管理 ESX 网络部件

在安装好 ESX 之后，系统会使用第一片网卡（vmnic0）创建第一个虚拟交换机（vSwitch0）。并且创建两个通信端口/组。第一个通信端口组虚拟机通信端口组 VM Network，用来让你创建 ESX 上的虚拟机，第二个当然是最重要的主控台通信端口。我们也可以自行创建更多的不同通信端口/通信端口组来实现更多的功能。我们在这一节，就来看看不同的网络设备创建及使用.



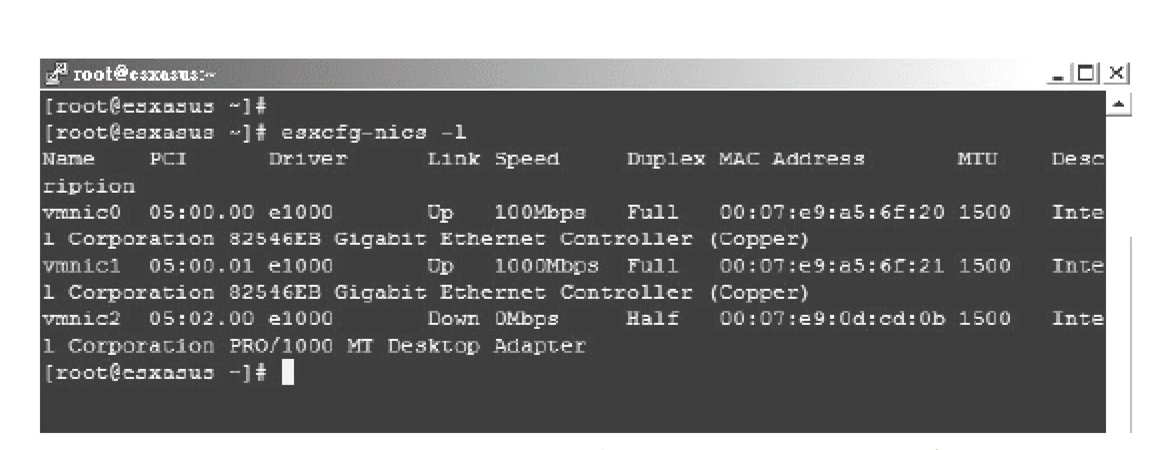
▲ 刚安装好 ESX 时会有一个默认的 vSwitch 在 vmnic0 之上

10.2.1 新建不同的网络设备

在 ESX 的物理机上默认使用 vmnic0 这片网卡，并且使用这片网卡上的 vSwitch0 为默认的交换机。主控台 vswif0 也是创建在这个交换机中。我们可以在这台机器上新建网卡、交换机和不同通信端口，并且让多张网卡进行备援等高级功能，接下来就来看看不同的设备如何在 ESX 上新建。

1. 新建网卡

首先必须确定这片物理网卡在 ESX 下可被系统辨识，只要你的网卡是 VMware HCL 中认可的，在 ESX 中安装都没有问题。我们可以先进入主控台键入 esxcfg-nics -l 的 Linux 命令来看看有没有找到网卡。系统会显示找得到的网卡，找到之后，就可以在 vSphere Client 中将这片网卡加入了。



▲ 可以从这里来确定是否安装成功　　▲ 在 vSphere Client 中也可以看到所有的网卡

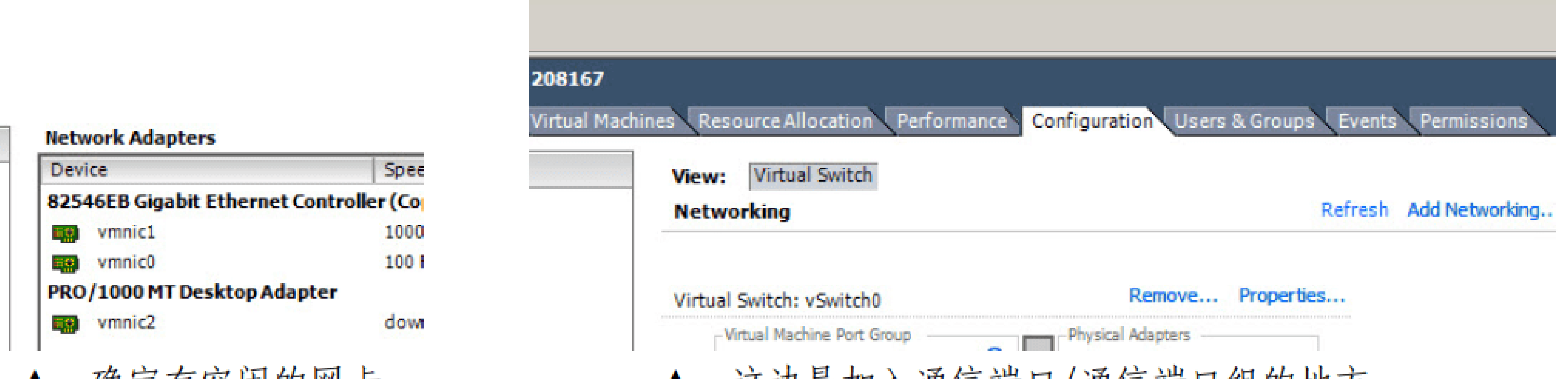
2. 新建虚拟交换机

在 ESX 中，虚拟交换机的新建必须搭配通信端口/通信端口组的创建，我们是无法创建空的虚拟交换机，一定要有一个服务需要创建时才能创建虚拟交换机，因此新建交换机就是新建通信端口/通信端口组。在 ESX 中，网络的单位就是虚拟交换机，因此我们可以针对不同的网络来配置交换机，然后再在交换机中新建不同的通信端口/通信端口组来落实服务的配置。我们就来看看新建虚拟交换机的方式。

▶ 新建虚拟交换机

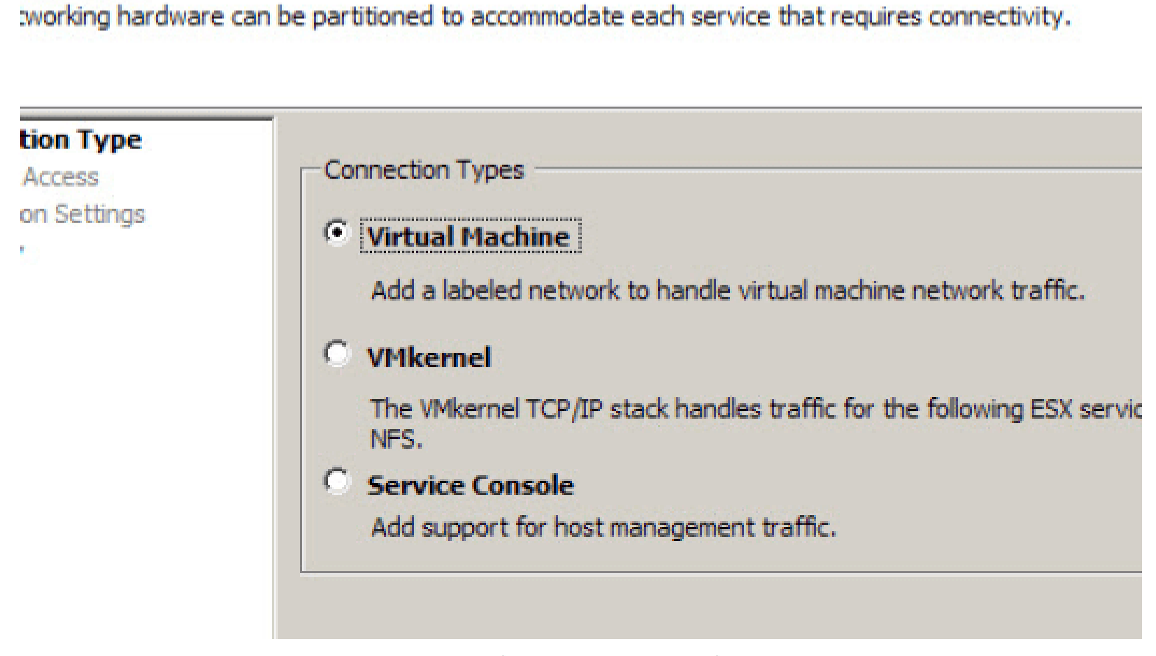
1．首先必须定有空闲的网卡可以使用。

2．进入 vSphere Client，并且选择 Configuration/ Networking 的 Add networking 选项。



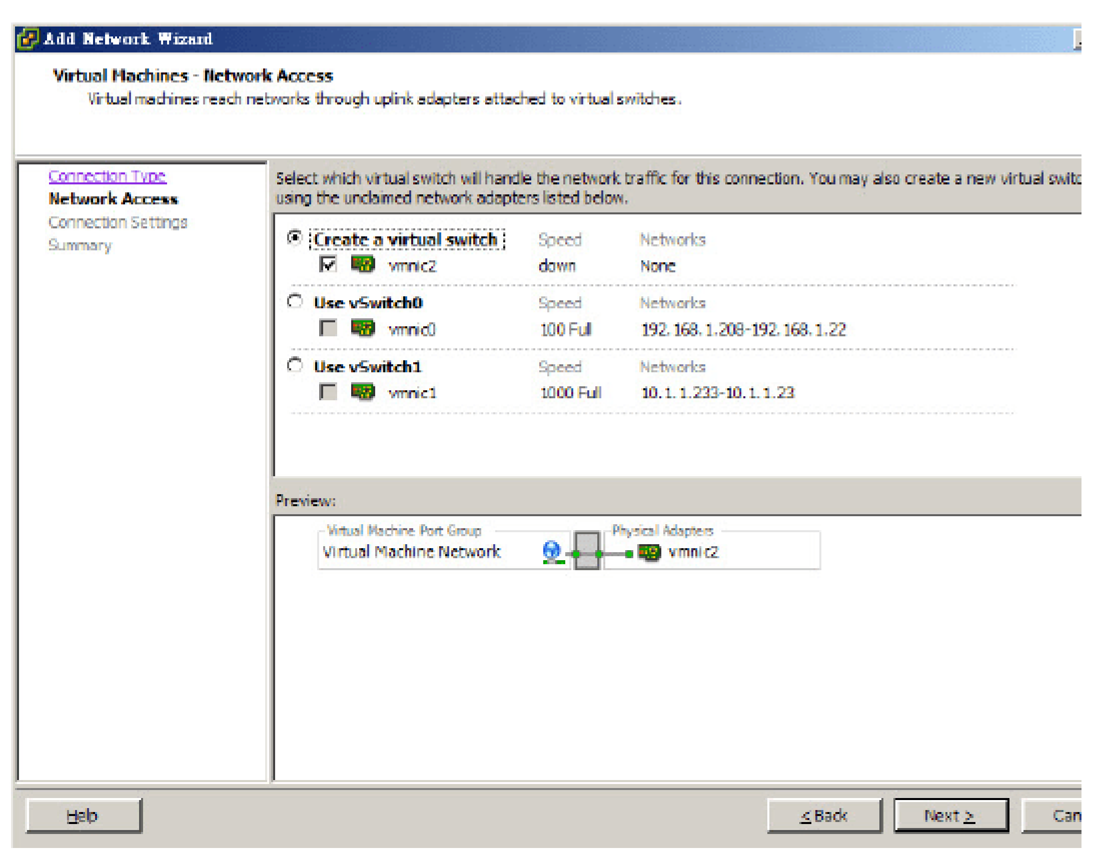
▲ 确定有空闲的网卡　　▲ 这边是加入通信端口/通信端口组的地方

3．接下来要选择使用的服务种类，在这里我们先选择给虚拟机专用的 Virtual Machine。单击 Next 按钮继续。



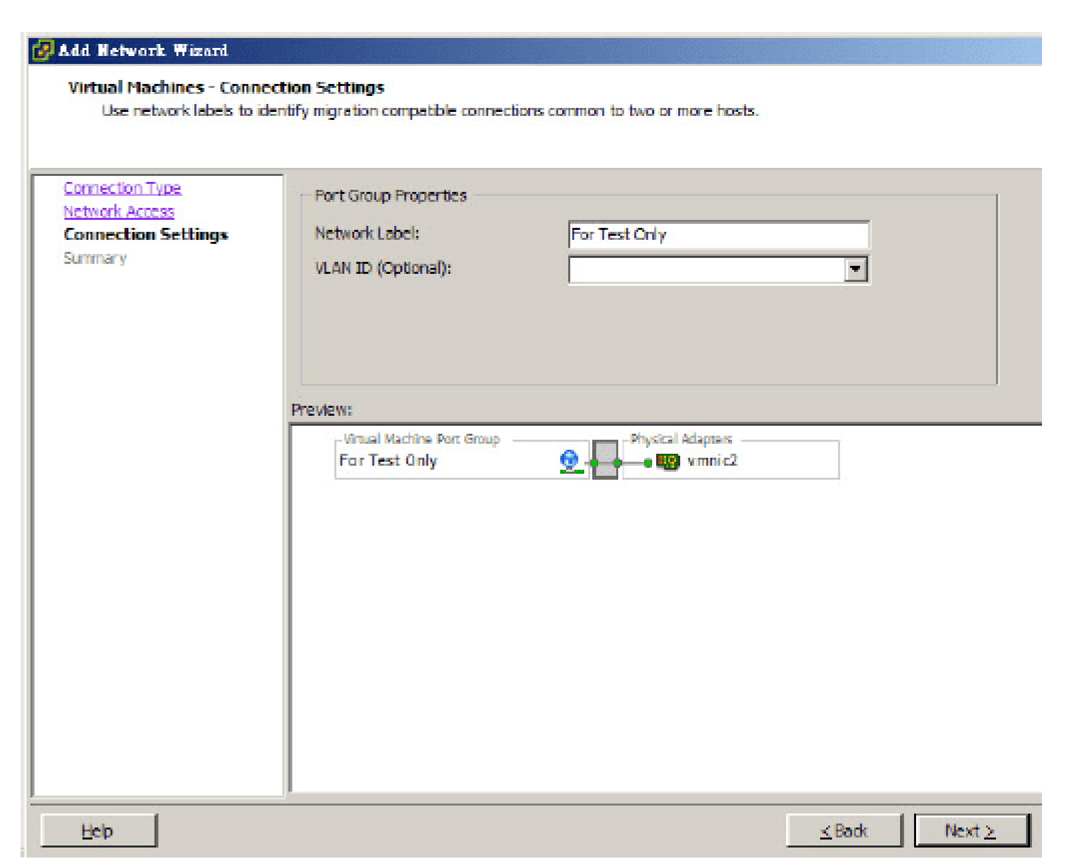
▲ 选择通信端口/通信端口组的种类

4．此时要决定我们要安装的 vSwitch 要放在哪一个网卡上。系统会默认使用空着的网卡 vmnic2。如果你用相同的网卡，则会将服务放在原来已经创建好的默认 vSwitch0。我们选择 vmnic2，单击 Next 按钮继续。



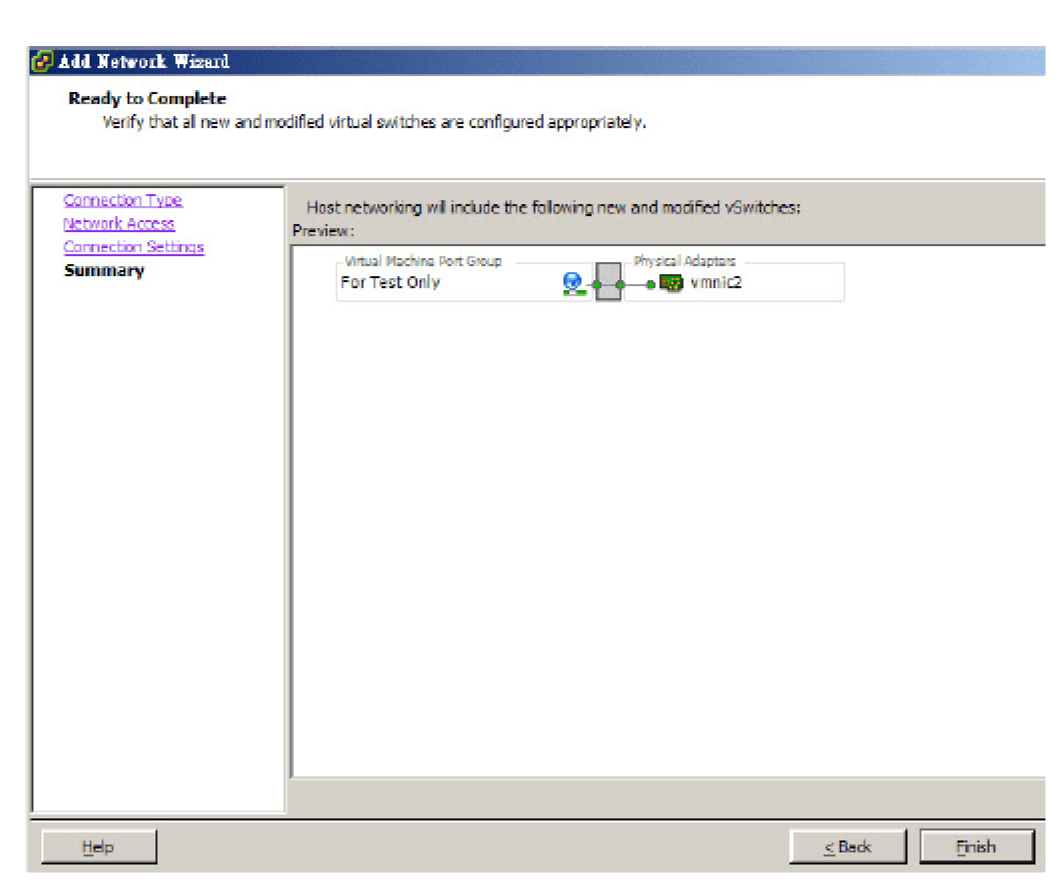
▲ 选择创建在新的网卡上

5．接下来键入这个 vSwitch 的名称，最好键入一个较好辨识的名称，如 For Test Only，键入完之后单击 Next 按钮继续。



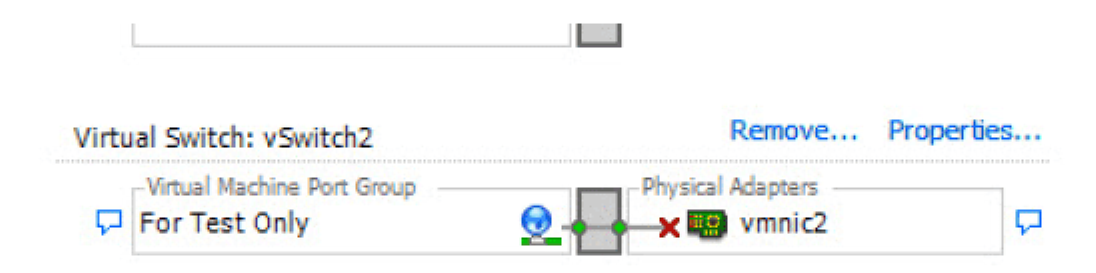
▲ 键入 vSwitch 的名称

6．在接下来的窗口中单击 Finish 按钮则会开始创建。



▲ 单击 Finish 按钮之后才会开始创建

7．之后在网络窗口中，就可以看到创建好的新的 vSwitch2 了。



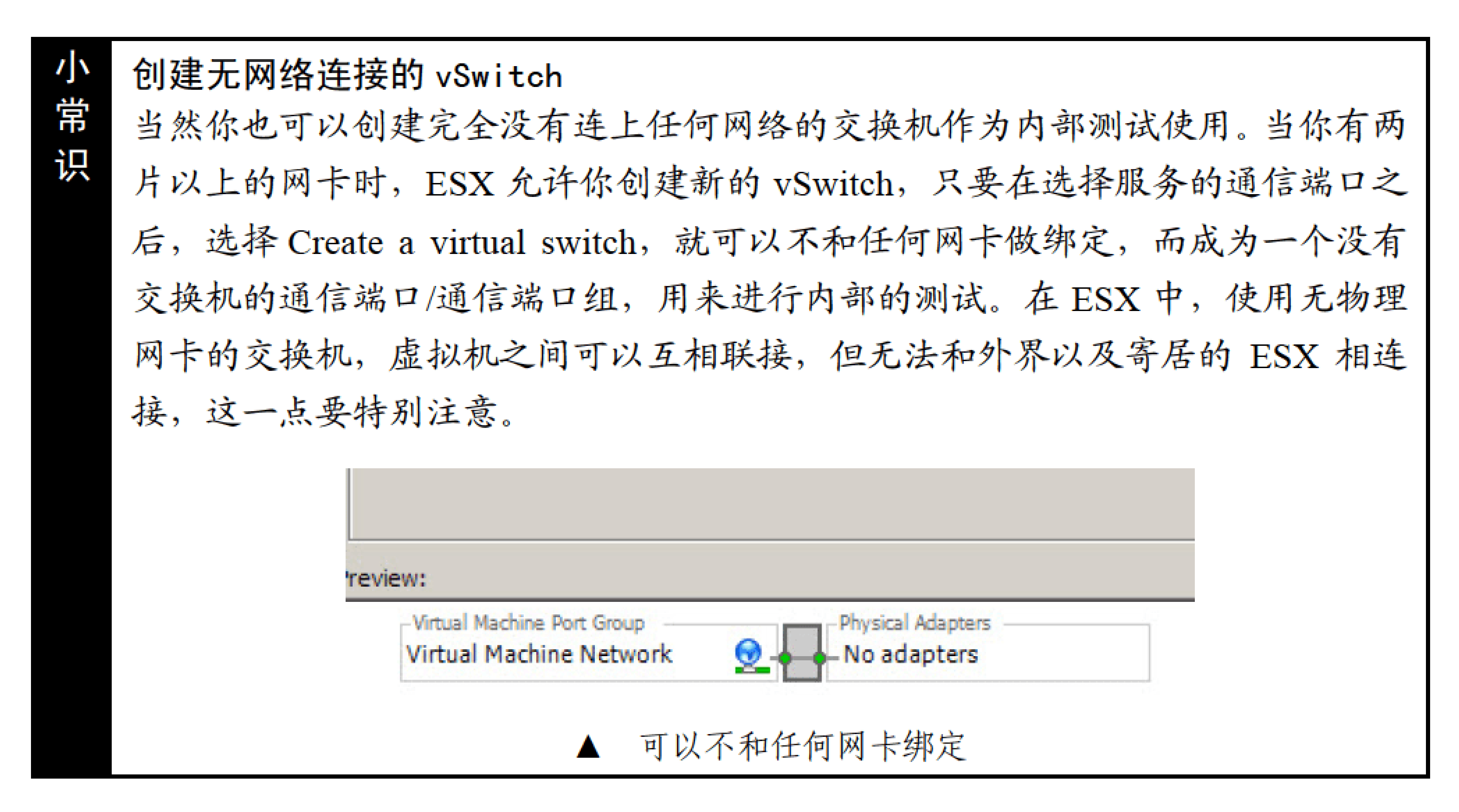
▲ 新创建的虚拟交换机

8．之后要创建主控台或是 VMKernel，都可以选择不同的 vSwitch。

小常识

创建无网络连接的 vSwitch

当然你也可以创建完全没有连上任何网络的交换机作为内部测试使用。当你有两片以上的网卡时，ESX 允许你创建新的 vSwitch，只要在选择服务的通信端口之后，选择 Create a virtual switch，就可以不和任何网卡做绑定，而成为一个没有交换机的通信端口/通信端口组，用来进行内部的测试。在 ESX 中，使用无物理网卡的交换机，虚拟机之间可以互相联接，但无法和外界以及寄居的 ESX 相连接，这一点要特别注意。



▲ 可以不和任何网卡绑定

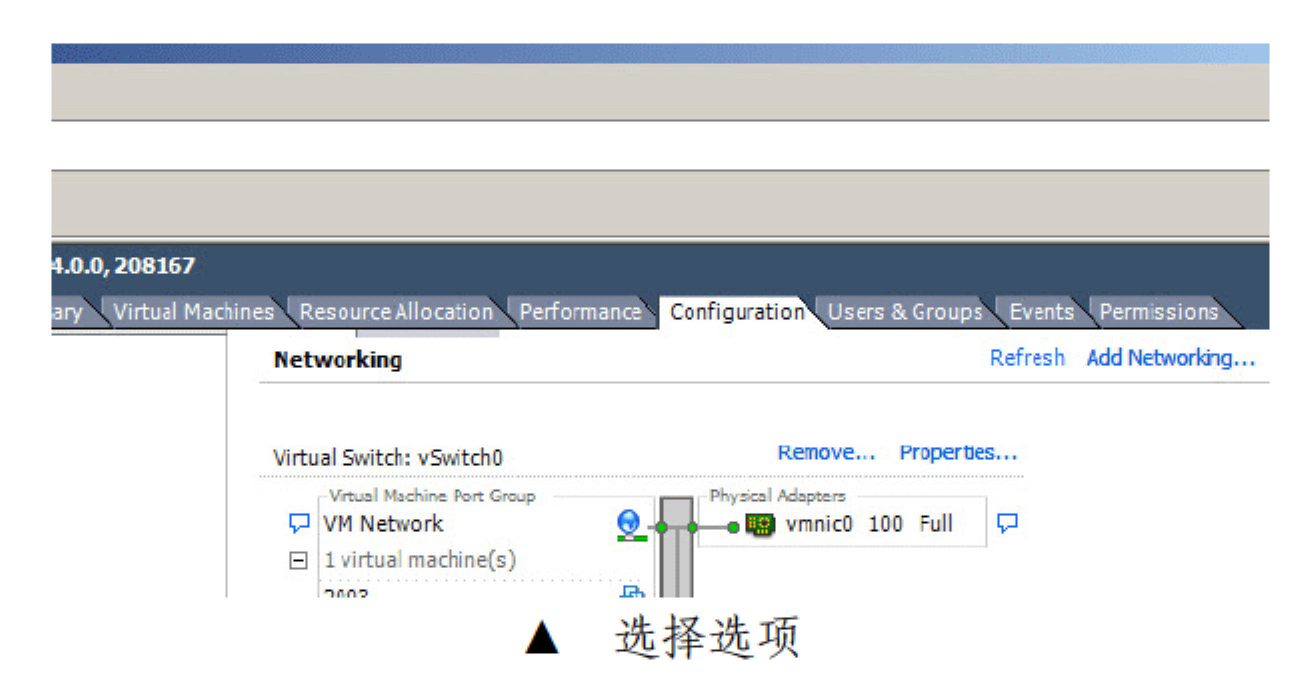
3. 创建新的通信端口/通信端口组

一般新建功能时，并发必须新建新的交换机，但你也可以将新的通信端口/通信端口组新建至原有的交换机上。新建的手段很简单，只要在新建通信端口/通信端口组之后，选择将功能放至旧的交换机上，就会将这个通信端口/通信端口组加入旧的交换机上了。

在 ESX 中，新建虚拟机通信端口和主控台/VMKernel 是不同的。由于虚拟机之间可能不需要和外网连接，形成一个封闭的网络，而这个封闭的网络中，会有一台出来担任 NAT/DHCP，因此 IP 是可以随便配置，只要虚拟机之间能沟通就行。但如果你要新建的通信端口/通信端口组是作为主控台或是 VMKernel 使用，系统就会要求你配置一个 IP 和外界沟通用，我们就以这个为例子来看看新建通信端口/通信端口组的手段。

▶ 新建 VMKernel/主控台

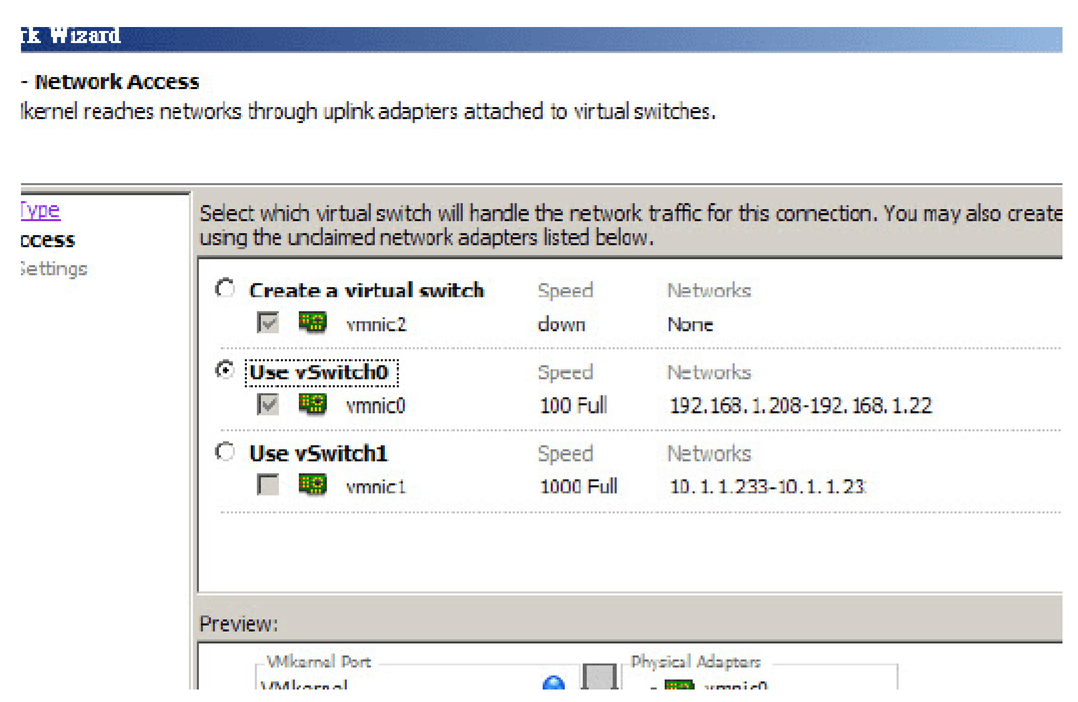
1．首先在 vSphere Client 的总控台中选择 Add networking 选项。



▲ 选择选项

2．选择 VMKernel。

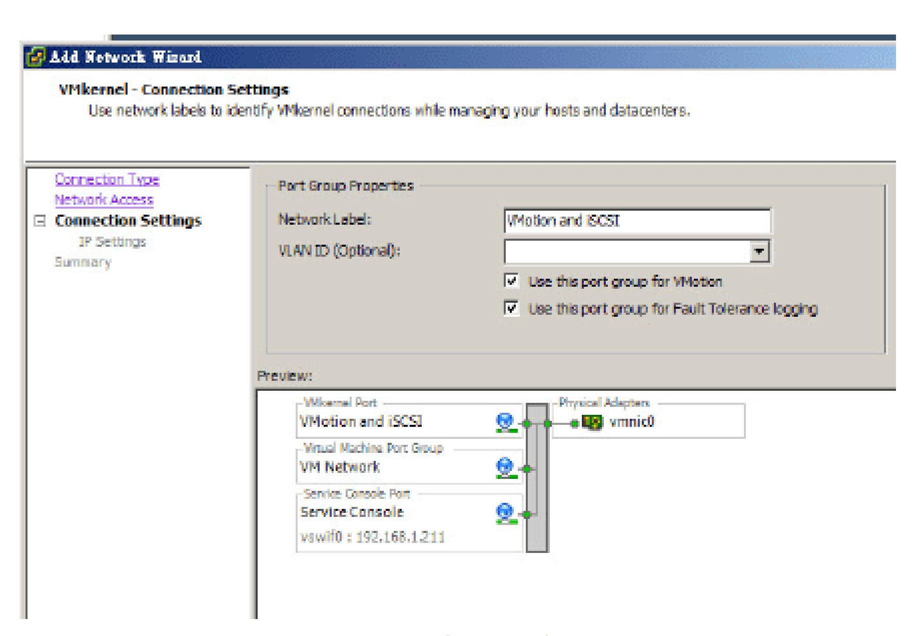
3．选择一个现存的 vSwitch，如 vSwitch0。



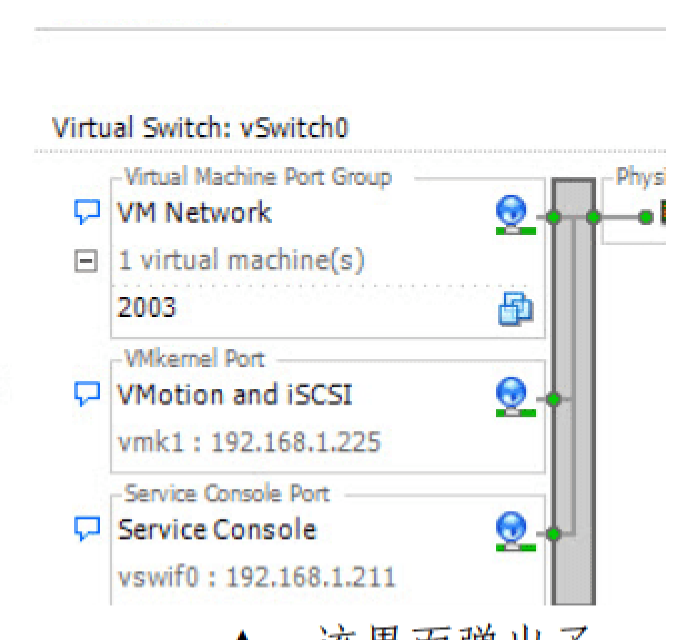
▲ 选择现存的 vSwitch

4．键入给定服务的名称，如 iSCSI，并且键入 IP 地址。如果你想让这个网络通信端口进行 VMotion，可以选中 VMotion 的选项。在 vCenter 环境中，如果机器有支持 FT，必须要选中 FT logging 选项才能激活 FT 功能。

5．最后单击 Finish 按钮继续。之后就可以看到该界面配置完毕了。



▲ 给定服务名称，并且选择 IP



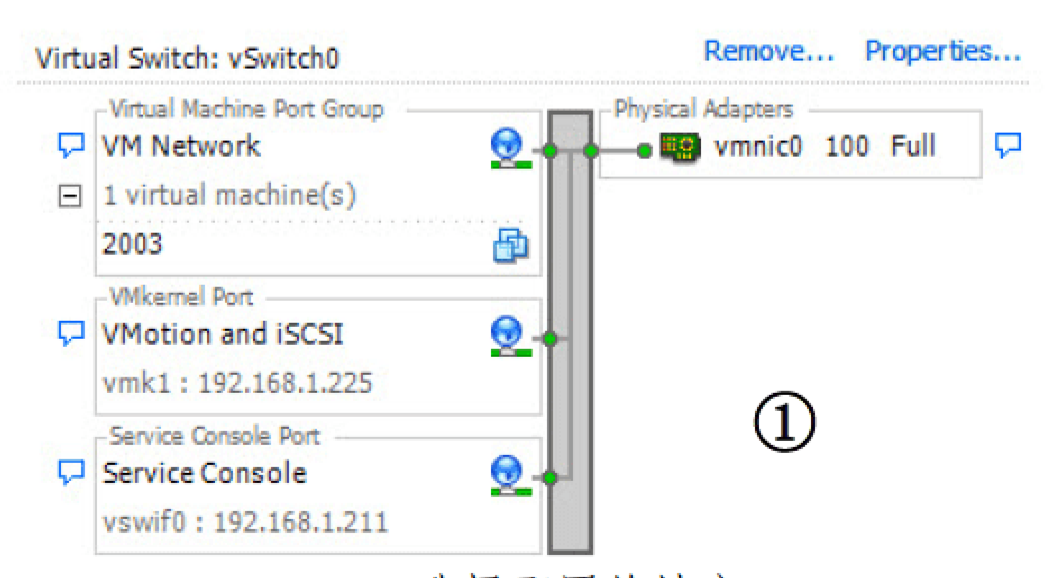
▲ 该界面弹出了

4．让一个 vSwitch 使用多片网卡

当我们在添加新的网络功能时，系统可以让我们选择多片网卡，但是对于已经安装好的 vSwitch0 怎么办呢？没问题，在你加入新的网卡之外，也可让原有的 vSwitch 享受多片网卡的安全及速度，下面就是详细的步骤。

▶ 添加 vSwitch 的多网卡绑定

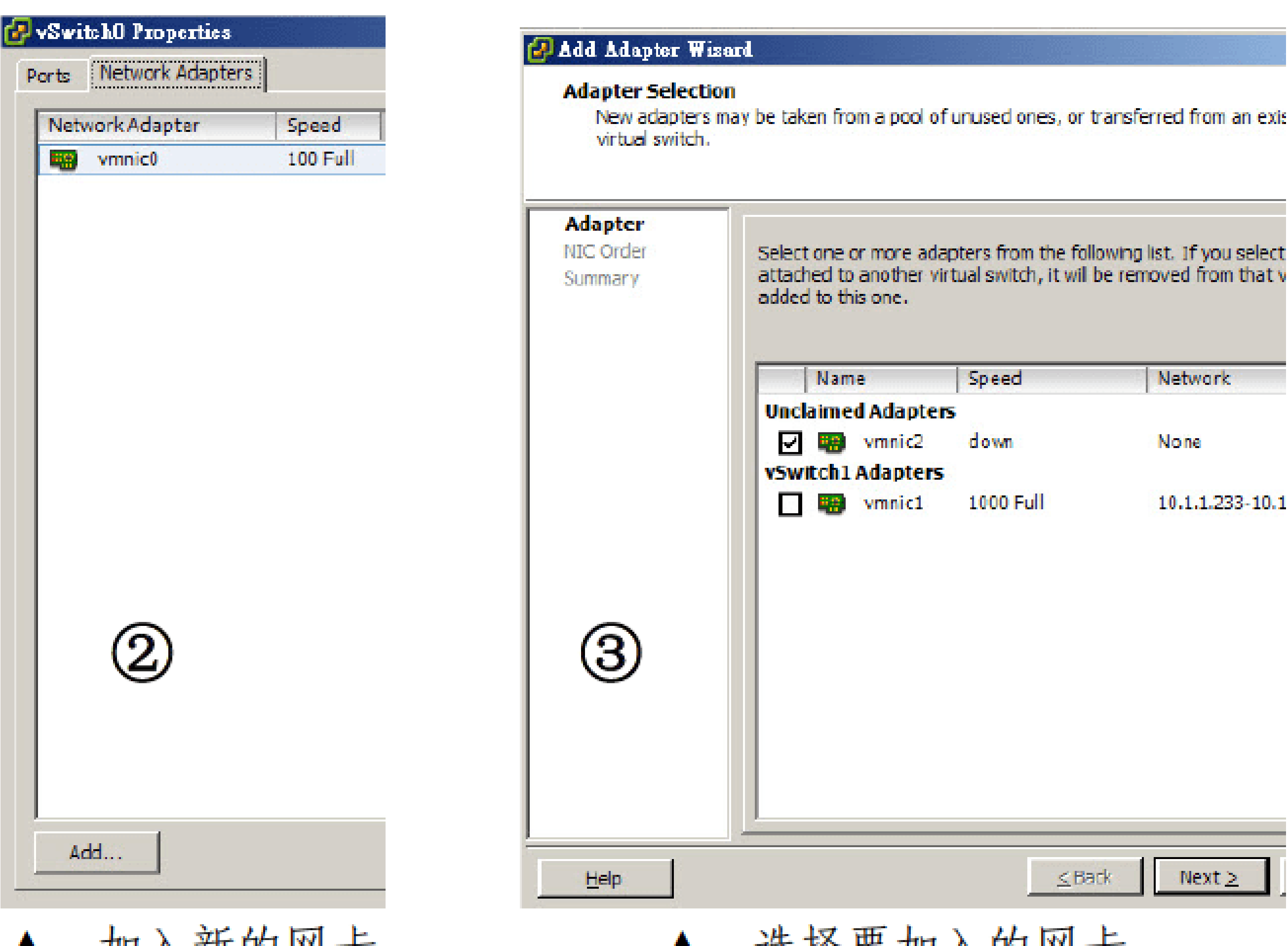
1．进入 vSphere Client 中，并且选择 Configuration/ Networking 选项，选择你要配置的 vSwitch，并且选择 Properties 选项。



▲ 选择配置的地方

2．选择 Network Adapters 选项卡，并且单击 Add 按钮。

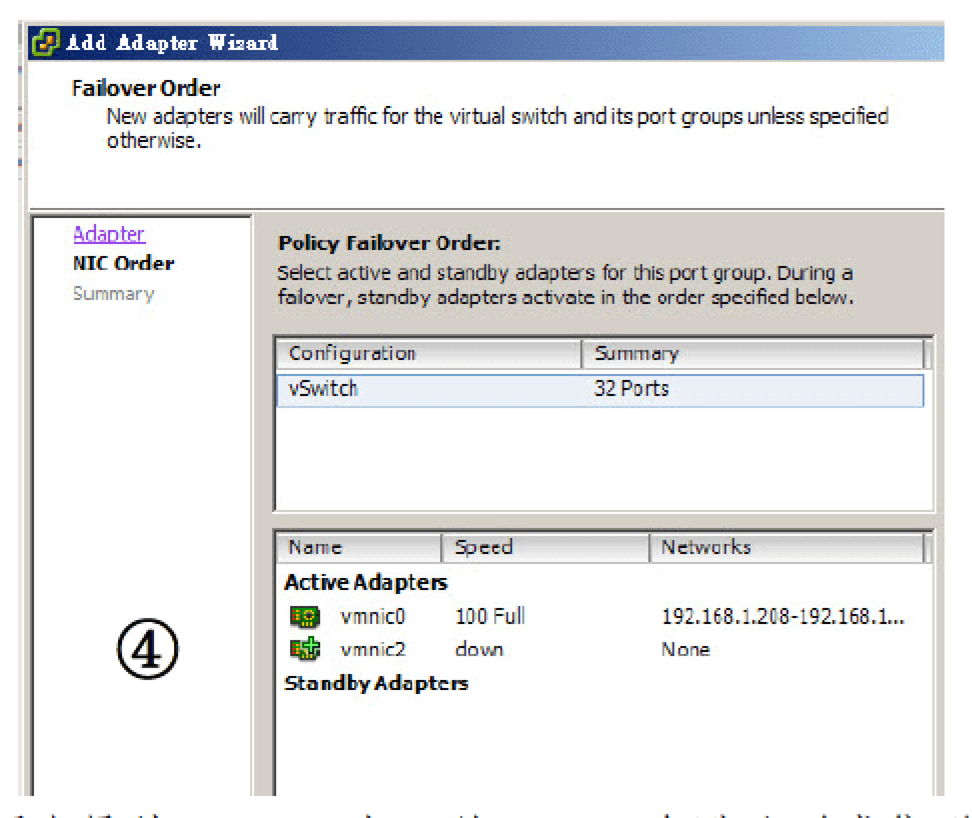
3．这是当前 ESX 抓到的所有网卡。将要选择的网卡打勾，并且单击 Next 按钮。

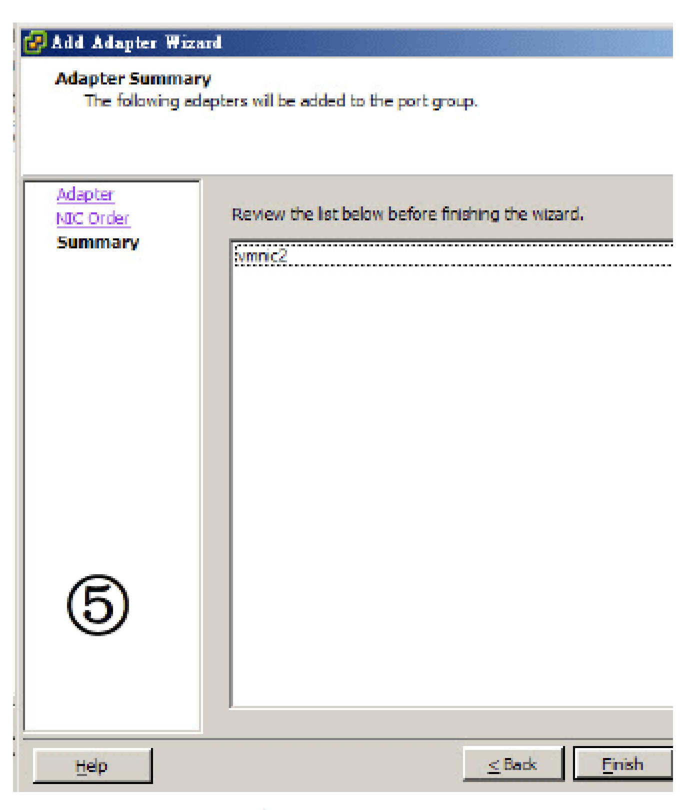


▲ 加入新的网卡　　　▲ 选择要加入的网卡

4．在拥有多片网卡之后，ESX 会默认让你激活 Failover Cluster 功能。我们当前先不管，直接单击 Next 按钮继续。

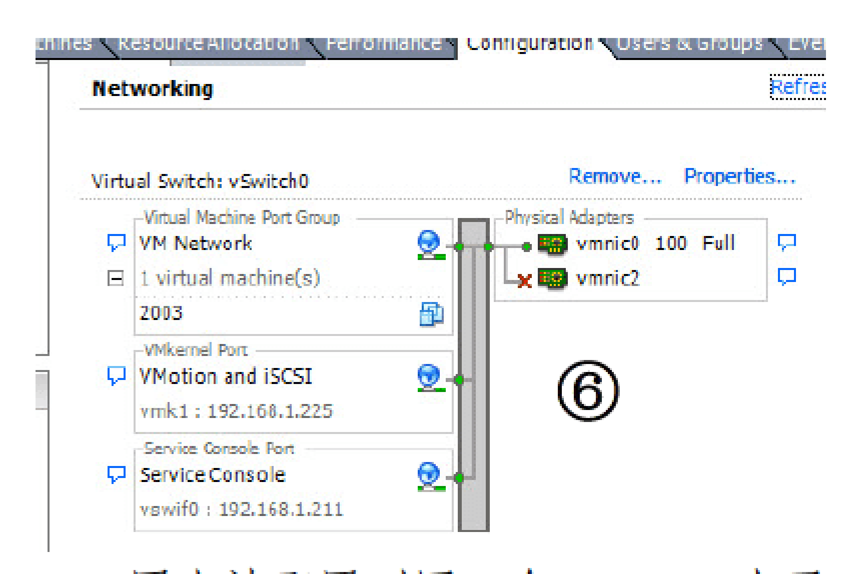
5．之后的画面为确定网卡的加入，我们直接单击 Finish 按钮即可。





▲ 由于网卡是从 vSwitch 加入的，ESX 会默认形成集群　　　▲ 确定网卡的加入

6．之后可以在网络的配置中看到新建的网卡。由于当前只有一个虚拟交换机，因此所有的网卡都会被配置在这个虚拟交换机上。一般来说，一个虚拟交换机使用多片物理网卡的情况主要还是防止单点失误的发生。



▲ 网卡被配置到同一个 vSwitch 上了

由于所有的网卡都被配置在同一个 vSwitch 上，因此我们无法再创建新的 vSwitch 了。

10.2.2 管理 ESX 的虚拟网络

当我们创建好所有的网络部件时，就可以将这些网络部件开始应用在虚拟机或管理上了。一般来说，ESX 自成一台的概率很低，通常一台 ESX 在 vSphere 架构中，会做出不同的虚拟交换机来应付不同的服务，并且也要照顾到防止单点失误、网络存储以及 VI 高级功能的使用，下面就是详细的说明。

1. 创建标准 vSphere 架构的 ESX 网络

在 vSphere 架构中，每一个 ESX 必须连上一台独立的存储，而每一台 ESX 上的虚拟机也必须和其他 ESX 上的虚拟机相连。并发我们也必须防止单点失误的发生。此外还要有主控台和存储设备、VMotion 等设计，因此一个匹配 vSphere 架构的 ESX 主机网络必须包括了下面几项。

▶ 匹配 vSphere 架构的 ESX 网络

（1）独立的主控台网卡虚拟交换机。

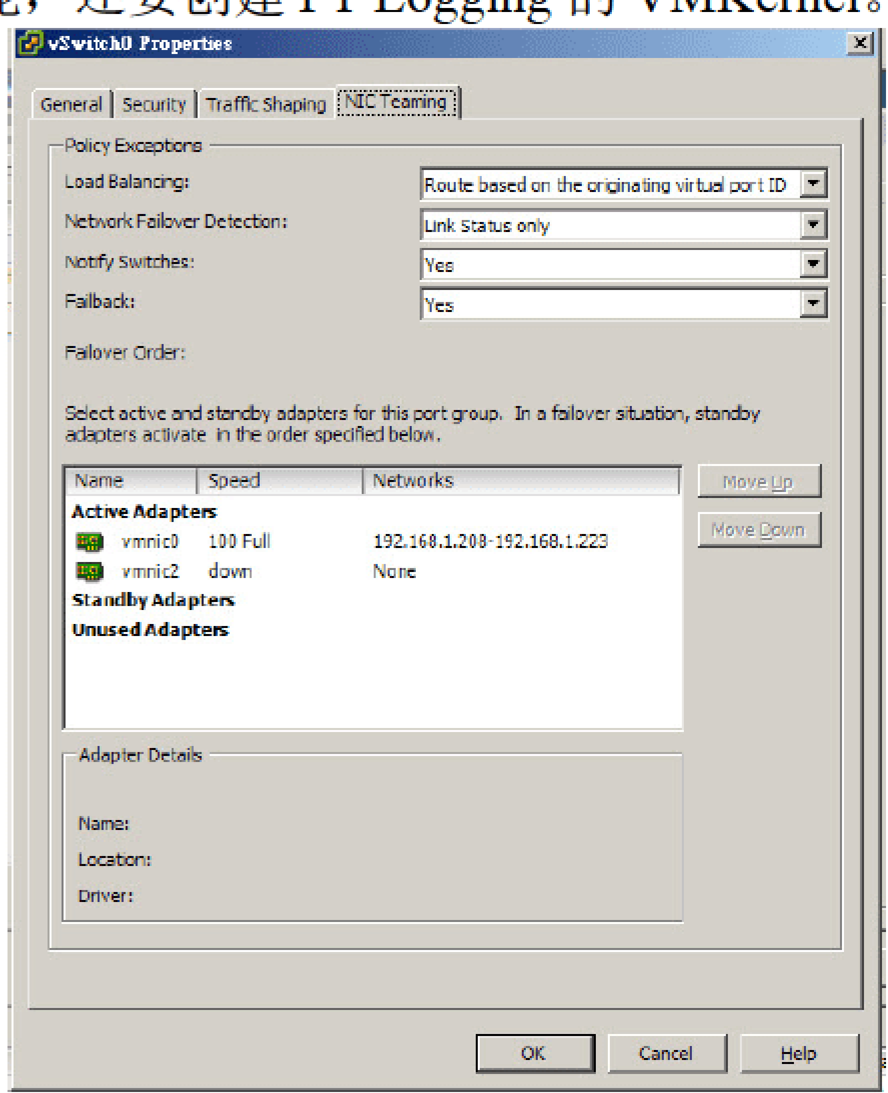
（2）具备防单点失误的多网卡虚拟机通信端口交换机。

（3）VMotion 使用的交换机。

（4）如果没有独立存储，必须有 iSCSI。

（5）测试用的交换机。

（6）如果要使用 FT 功能，还要创建 FT Logging 的 VMKernel。



▲ 标准 vSphere 架构的 ESX 网络，主要是生产环境的 NIC Team

一般来说，防止单点失误的多网卡虚拟交换机可以使用 NIC Team 功能，我们在设计时使用多网卡就可以达到这个目的。上图就是一个标准的 vSphere 架构 ESX 网络拓扑，读者们可以依照上一节新建网卡、虚拟交换机以及不同通信端口组的方式来创建作为练习。

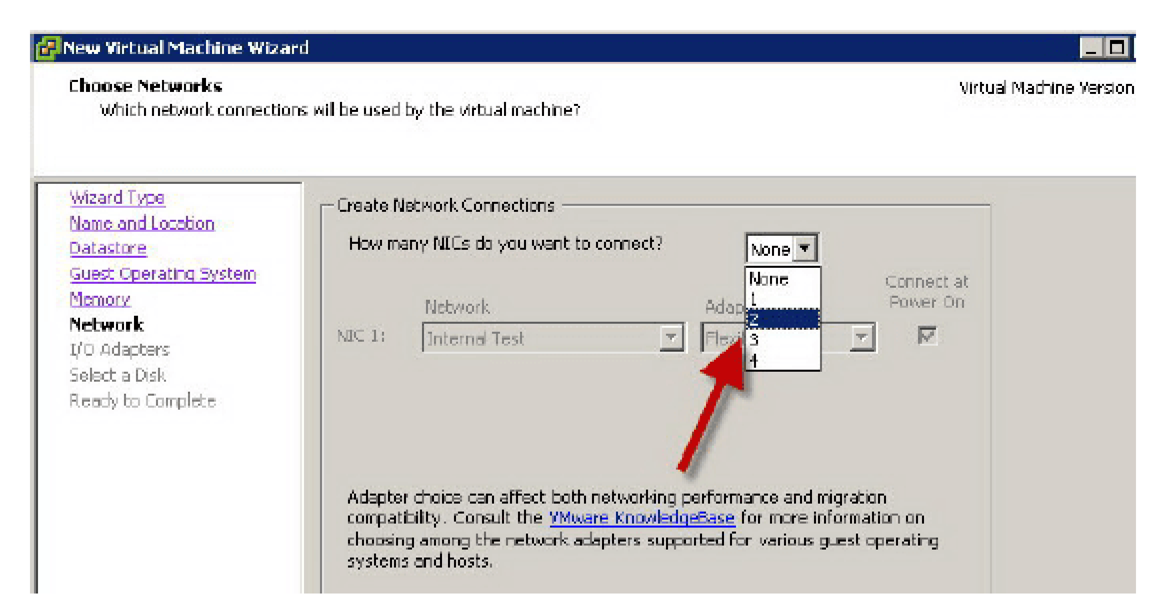
2. 把虚拟机连上虚拟交换机

当我们创建好不同的交换机时，在 ESX 上的虚拟机当然可以连上不同的交换机。但前提是连上的交换机必须具有虚拟机的通信端口组。如果我们在 ESX 中有配置多个不同的虚拟机通信端口组，那么就必须在配置虚拟机时多分配几片网卡，分别联机到不同的虚拟交换机上。举例来

说，我们创建一个新的 Windows Server 2003，在上述的例子中，就可以配置两片网卡，一片联机到 Production Environment，另一片联机到 Internal Test。此时这台 Windows Server 2003 可以连上两个不同的网域，就可以充当两个网域之间的路由器或是进行 looping 的任务了。我们就来看看给定的方式。

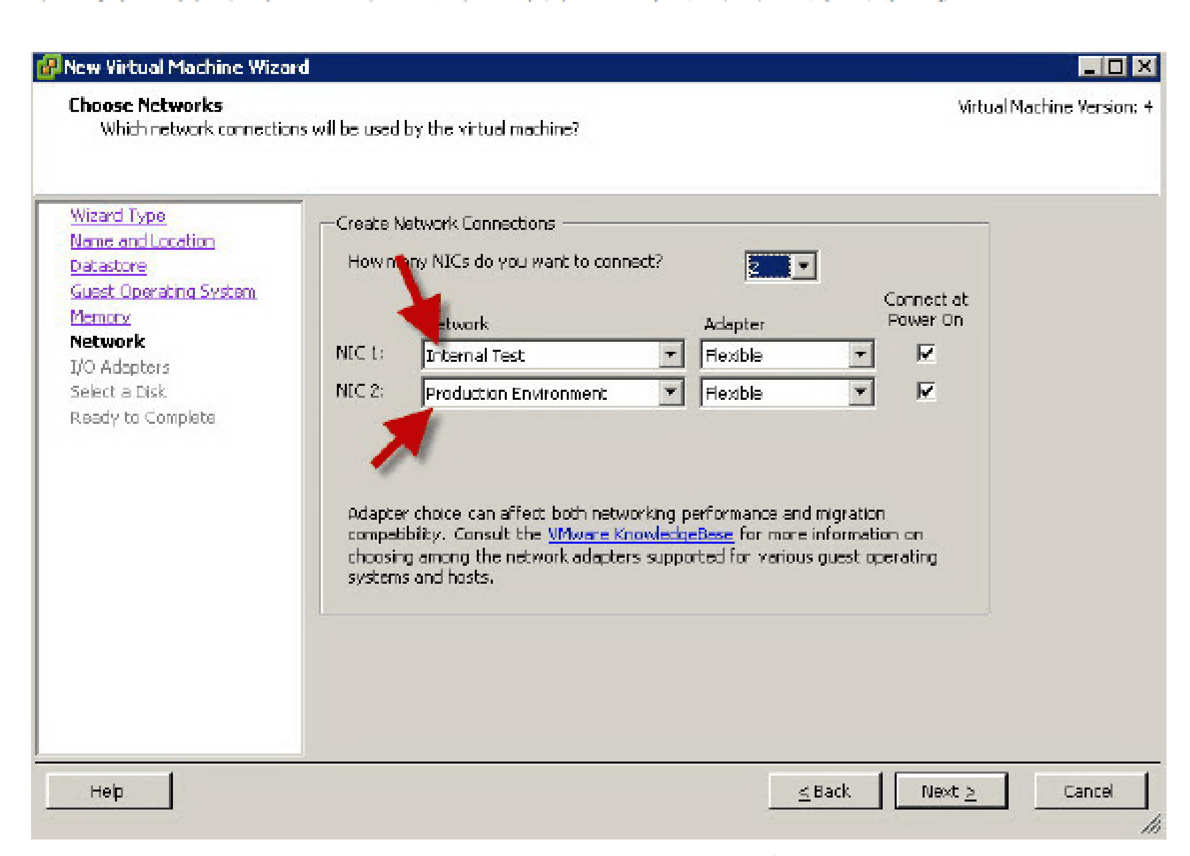
▶ **给定虚拟机联机到不同的交换机**

1．在创建虚拟机时，我们会进入网卡的步骤。此时别忘了将网卡的数量配置成 2，并且把不同的网卡配置到不同的交换机上。



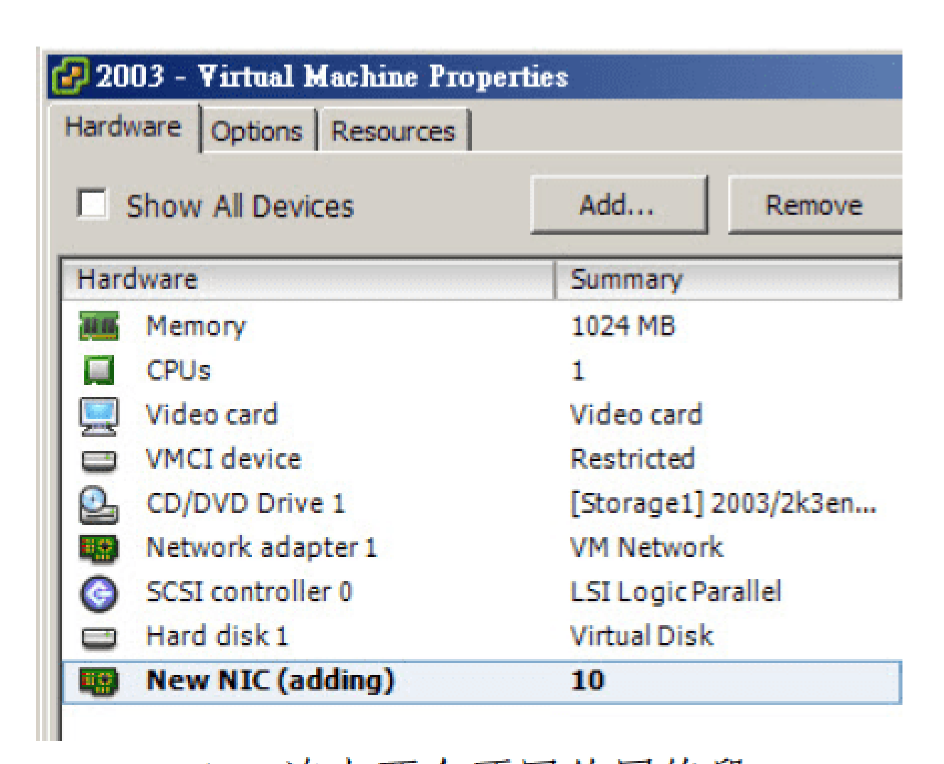
▲ 配置两个网卡

2．此时我们可以看到两个网卡可以给定到不同的交换机上。



▲ 连上两个交换机

3．在引导安装操作系统，并且安装了 VMware Tools 之后，两个网卡都分别连上不同的网段了。



▲ 连上两个不同的网络段

结 语

本章是 ESX/ESXi 网络最基本的部分，所有 vSphere 的高级功能，都和 ESX/ESXi 的网络功能有极大的关系，因此理解 ESX/ESXi 的网络，是真正操作 vSphere 环境的基础，读者们一定要将本章的观念和操作熟练，在管理 vSphere 环境时才能真正驾轻就熟。

读书笔记

第 11 章 完全征服 vSphere 上的存储理论及实践

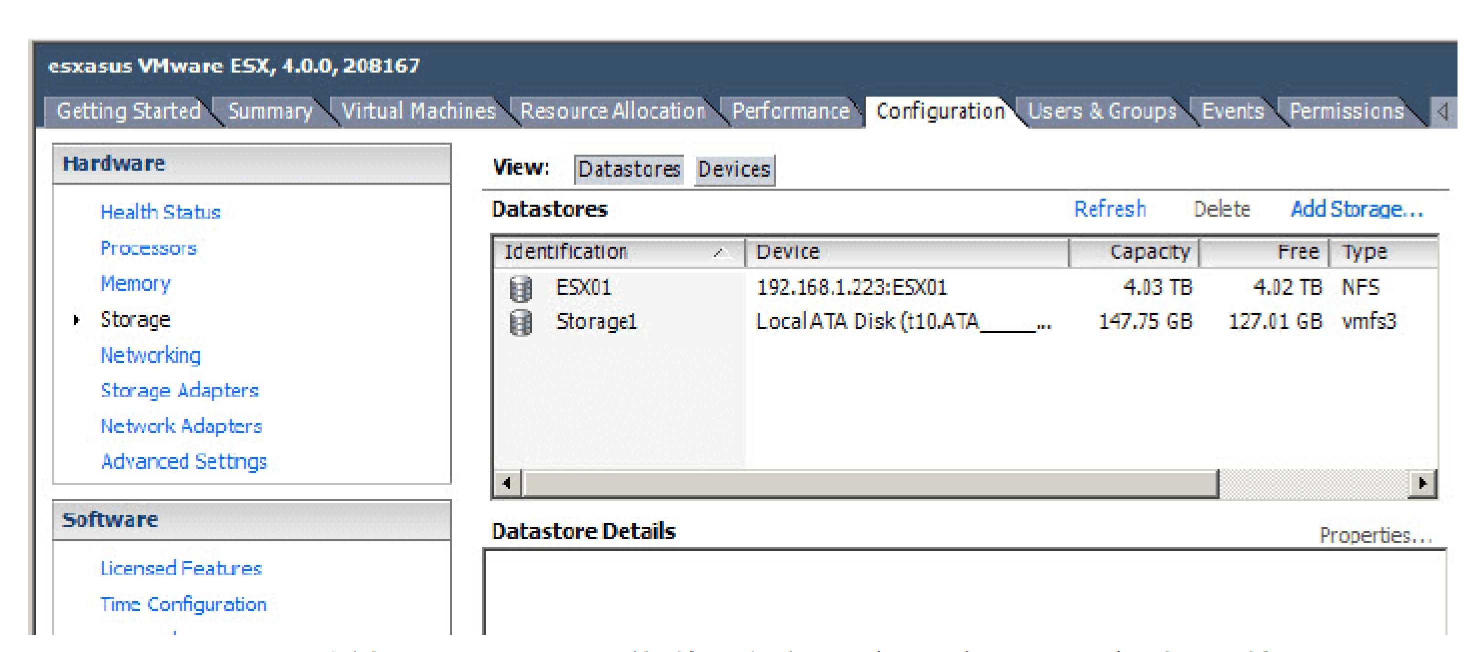
关键词：

- 理解 ESX/ESXi 的存储界面卡
- 理解 ESX/ESXi 的网络存储界面卡
- 理解独立存储和内部存储
- 理解什么是 iSCSI
- 理解什么是 NFS
- 理解什么是 Datastore
- 突破单个 LUN/硬盘 2TB 的制约

存储设备是虚拟化的基础，没有独立的存储设备，用户只能做到单台 ESX/ESXi 上的多台 VM，这对公司的 IT 虚拟化一点帮助也没有。但在加入了独立的存储设备之后，整个 vSphere 环境俨然落实。所有的 VM 转移功能、HA 功能、DRS 功能，都得靠独立存储设备才能落实。我们在这一章，就来完全理解 vSphere 中的存储设备，如果对存储不理解的读者，可以参阅第 4、5 两章的说明。

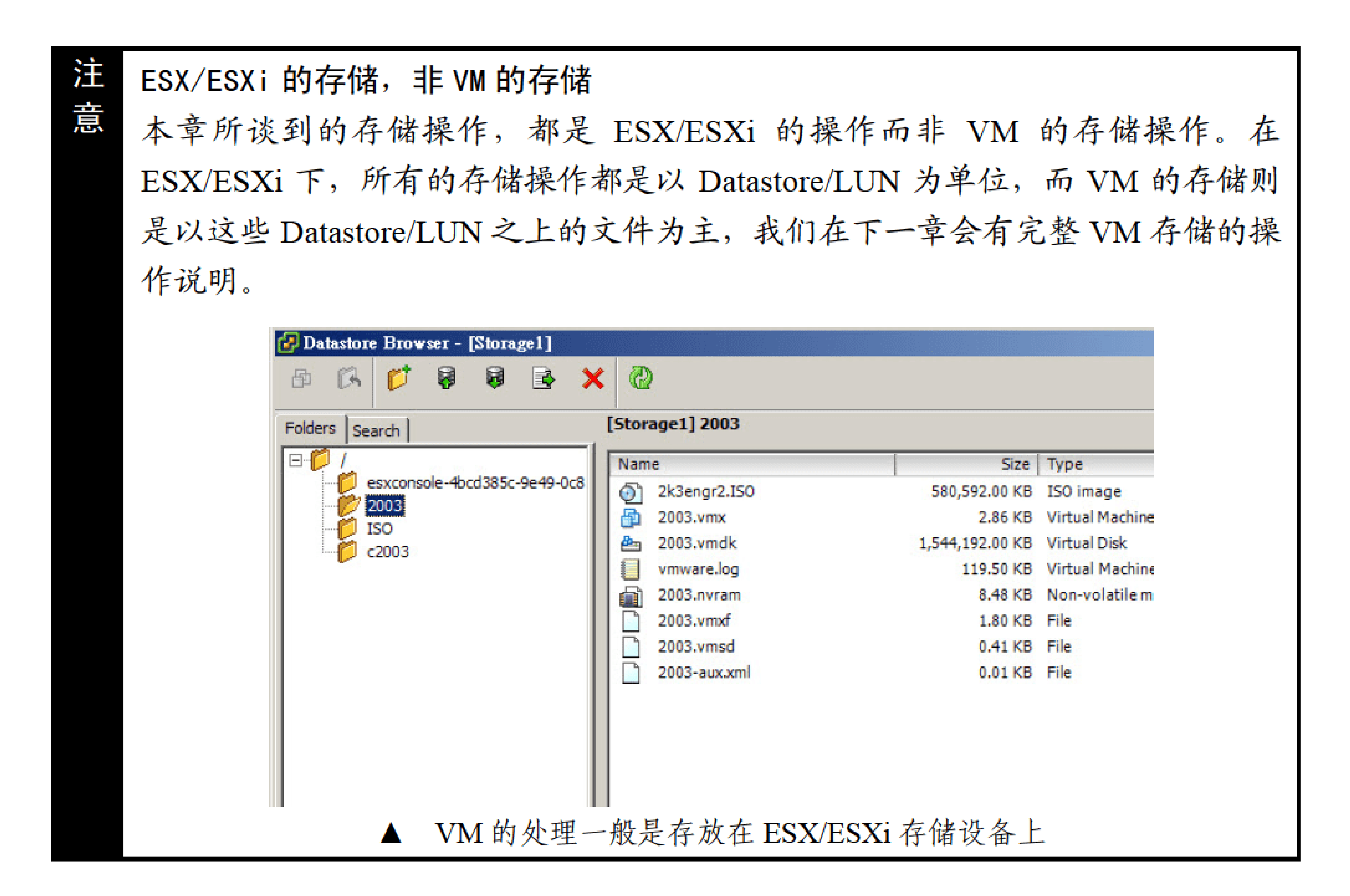
11.1 理解 ESX/ESXi 的存储设备

在 ESX/ESXi 中，存储设备的主要功用就是置放 VM 及其硬盘文件，包括了 VM 本身的配置文件.vmx、虚拟磁盘 VMDK，以及快照的 Redo 文件。在一台 ESX/ESXi 中，最简单的存储设备就是本地的硬盘，但最重要的还是独立于 ESX/ESXi 服务器之外的独立存储设备。ESX/ESXi 提供了一个统一界面来管理内部及外部的存储设备。



▲ 只要是 ESX/ESXi 能找到的设备，都可以在这里管理

> 注意
>
> **ESX/ESXi 的存储，非 VM 的存储**
>
> 本章所谈到的存储操作，都是 ESX/ESXi 的操作而非 VM 的存储操作。在 ESX/ESXi 下，所有的存储操作都是以 Datastore/LUN 为单位，而 VM 的存储则是以这些 Datastore/LUN 之上的文件为主，我们在下一章会有完整 VM 存储的操作说明。

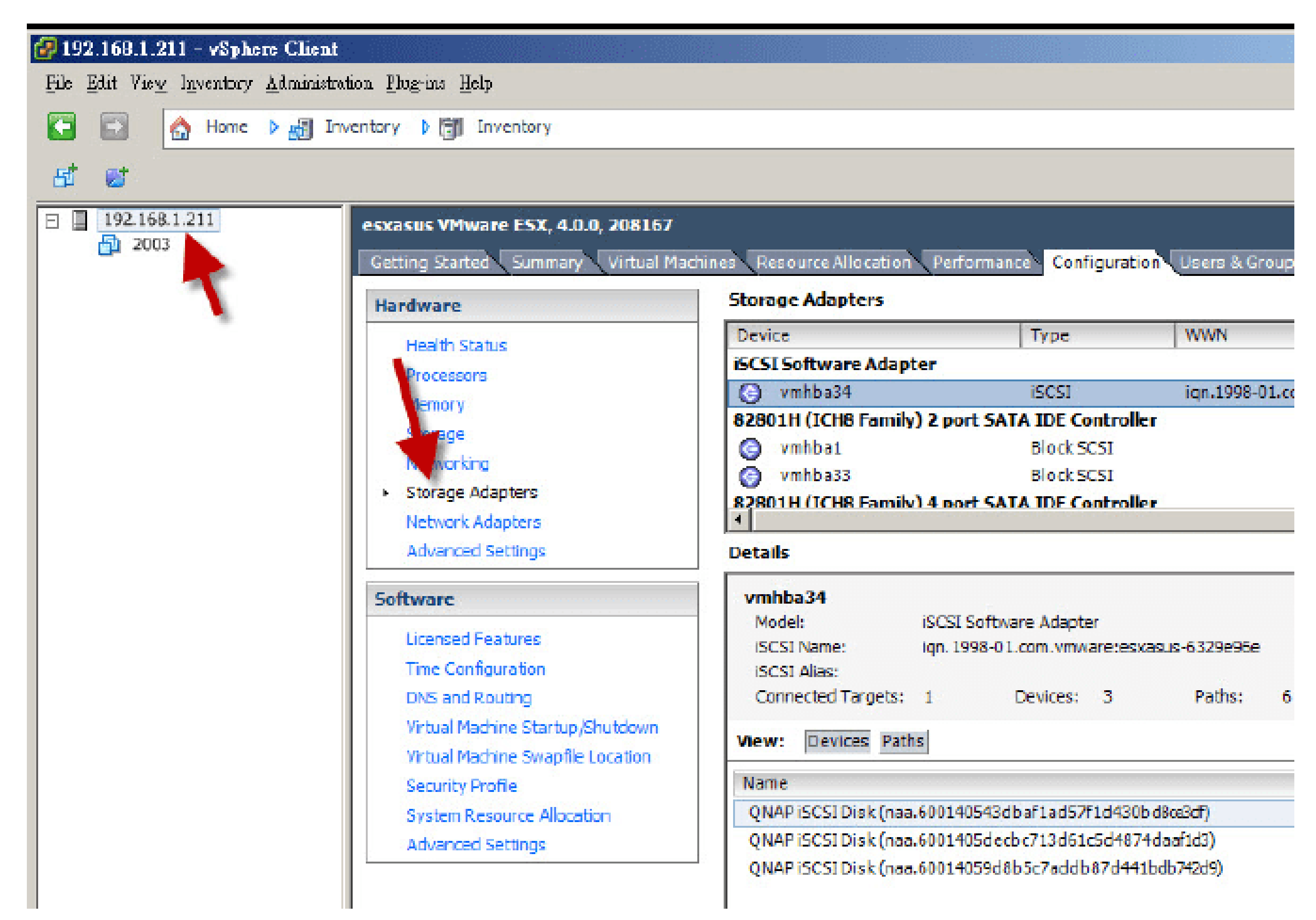


▲ VM 的处理一般是存放在 ESX/ESXi 存储设备上

11.1.1 理解 ESX/ESXi 的网络存储原理：存储界面卡

在进入 ESX/ESXi 主机之后，进入 Configuration 的选项卡，可以见到 Storage Adapter 的选项。这个选项包含了这台 ESX/ESXi 主机下能辨识的所有存储界面卡。存储界面卡就像 PC 上的硬盘 IDE 或 SATA 界面一样，在 PC 上是以南桥芯片来总控，但是在大部分的服务器上，则是由一片单独的卡来管理。这片卡通常都有 RAID 功能，并且可以创建不同的 Volumes。在 ESX/ESXi 中，

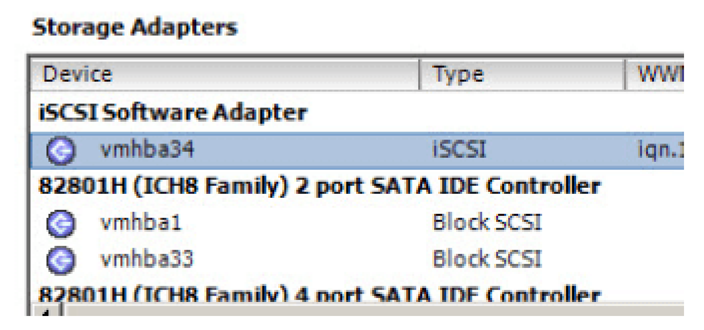
存储界面卡分为物理界面卡（如本地的 SATA、硬件的 iSCSI 卡或是 FC SAN 的 HBA）以及软件类界面卡（将网卡类比成 iSCSI 卡）。



▲ 这里是 ESX/ESXi 管理存储界面卡的地方

1. 虚拟 HBA 卡：vmhba

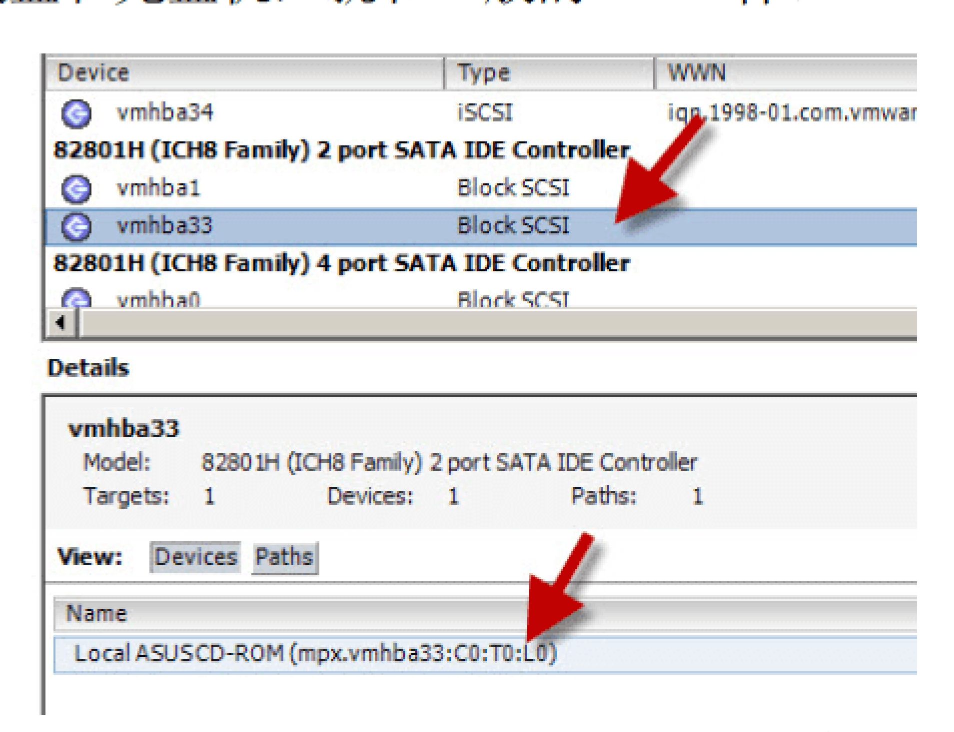
HBA 是专属的存储设备连接卡，但是在 ESX/ESXi 中，只要连接到 Datastore 的界面都被称为 HBA，不管是本地、HBA 或是 iSCSI 界面卡，在 ESX/ESXi 下都会被命名为 vmhba。在安装好 ESX/ESXi 时，连接到本地磁盘的第一片 HBA 被称为 vmhba0，以后再加上的 iSCSI、FC SAN 或是 NFS 则依加入的顺序称为 vmhba1、vmhba2。由于 vmhba 下可以连接多个设备，因此会以 vmhba0:0:0:3 的格式来给定同一个 HBA 卡上的不同存储设备。



▲ 在 ESX/ESXi 下，不管是什么界面卡都被命名为 vmhba

2. 本地的界面卡

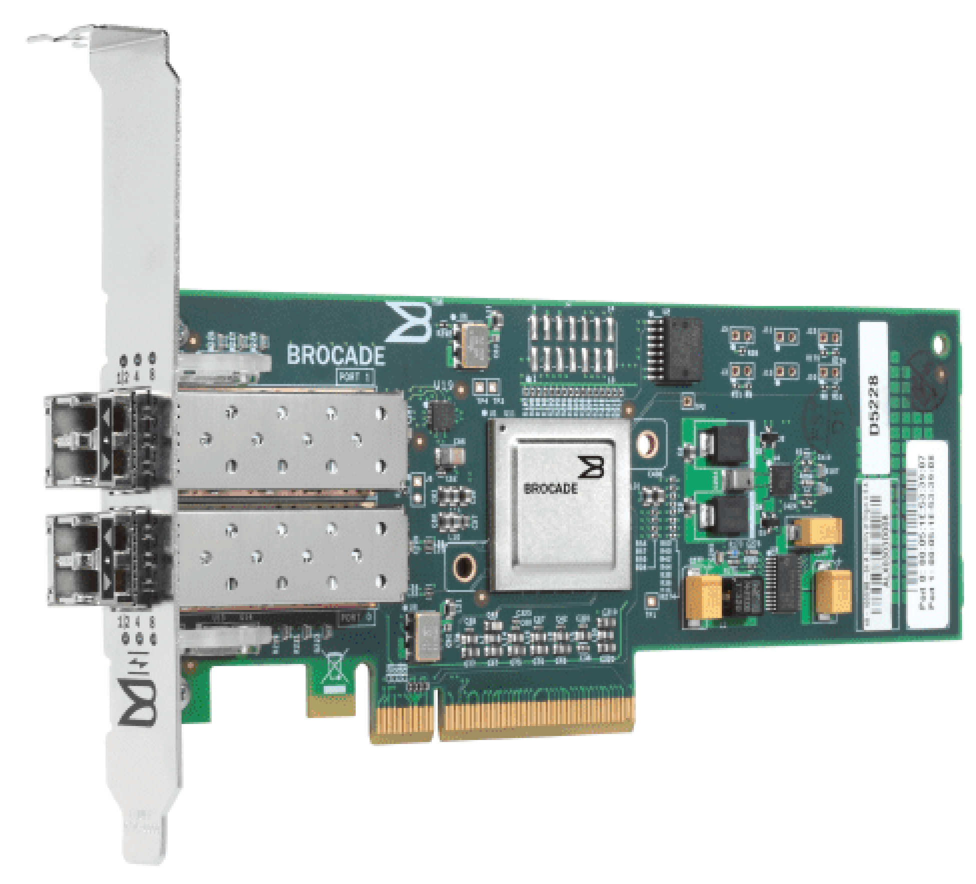
如果主机的界面卡是匹配 vSphere HCL 的话，ESX/ESXi 下可以看到这片界面卡。本地界面卡通常会显示连接的本地硬盘和光盘机，就和一般的 PC 一样。



▲ 本地的界面卡，下面会给出设备

3. HBA 界面卡

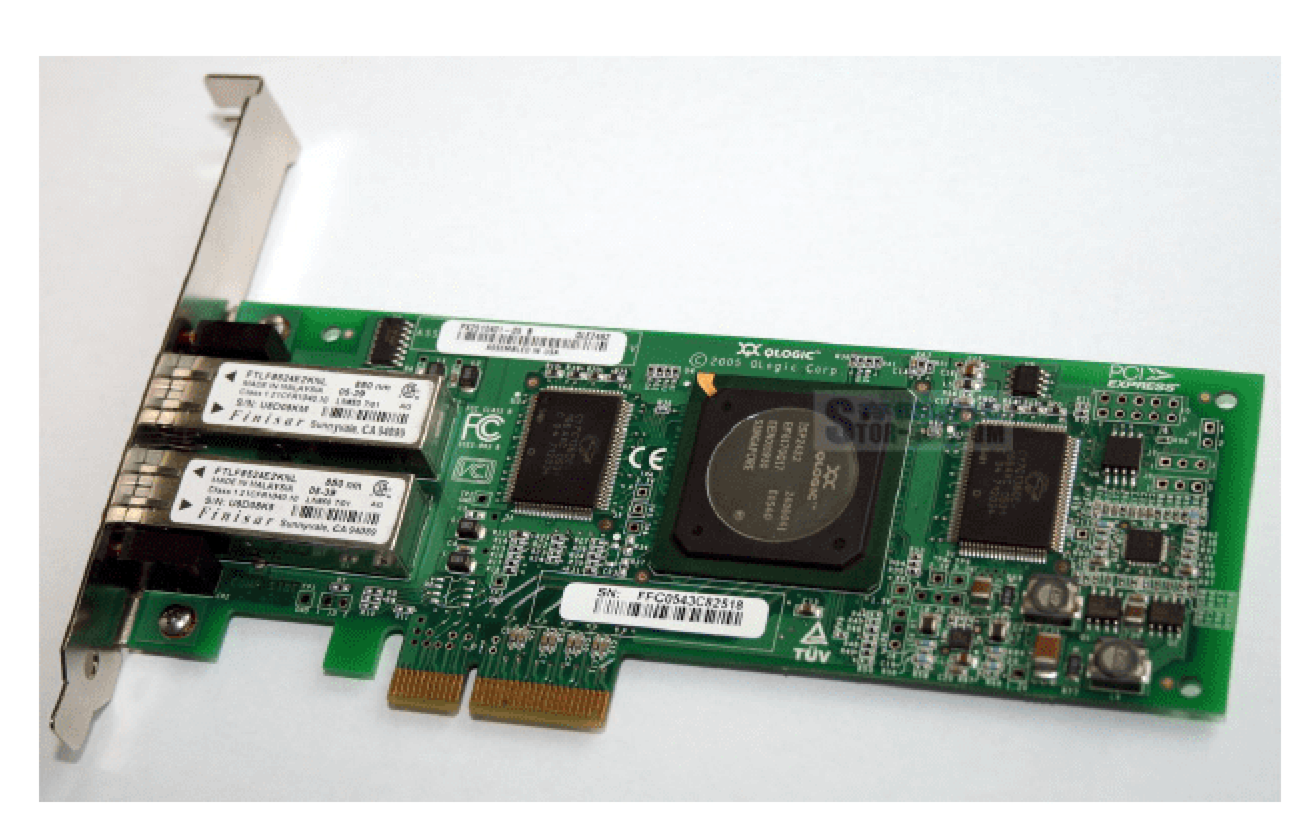
就是真正的 HBA，通常指连到外部光缆存储的 FC SAN 卡或是具有硬件功能的 iSCSI 连接卡。



▲ 真正的 FC SAN 卡成本较高

4. iSCSI Software Adapter

软件 iSCSI 本来就是将 SCSI 命令用 IP 报文加壳提交，因此网卡本身就是 iSCSI 卡。但由于也有硬件的 iSCSI 卡，因此在 ESX/ESXi 中，称为 Software Adapter。本章的示例就是以软件方式来演示。



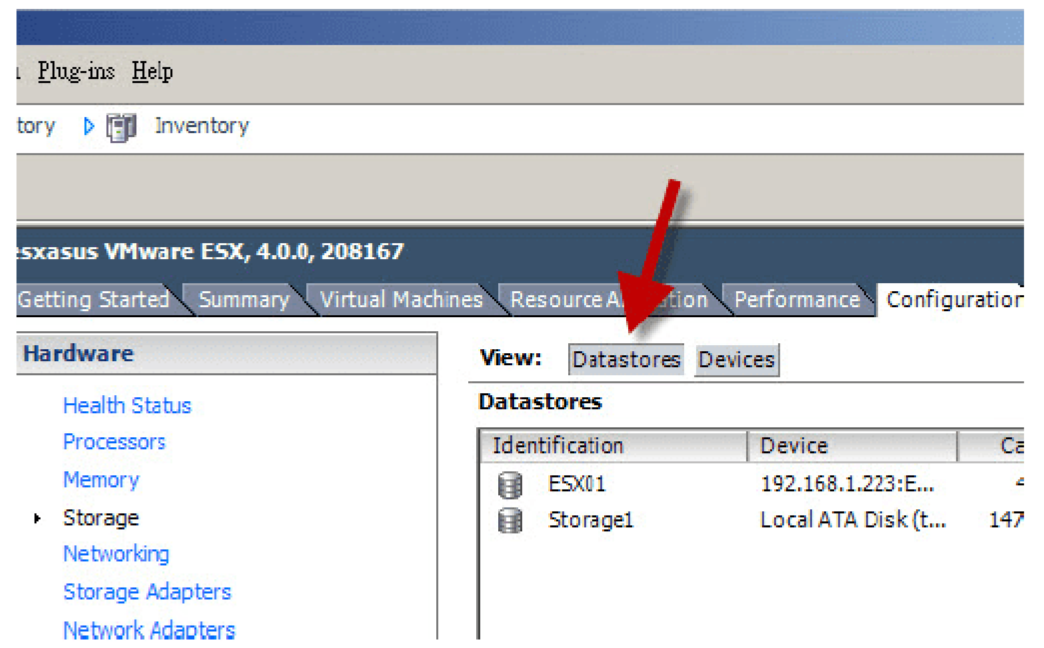
▲ iSCSI 是当前最流行的存储卡

11.1.2 理解 ESX/ESXi 的存储原理：存储设备

在理解界面卡之后，界面卡上连接的设备就是存储设备了。在 ESX/ESXi 下，可以在 Configuration 的选项卡下选择 Storage 选项。这里就是管理 ESX/ESXi 所有存储设备的地方。

1. Datastore：数据仓库

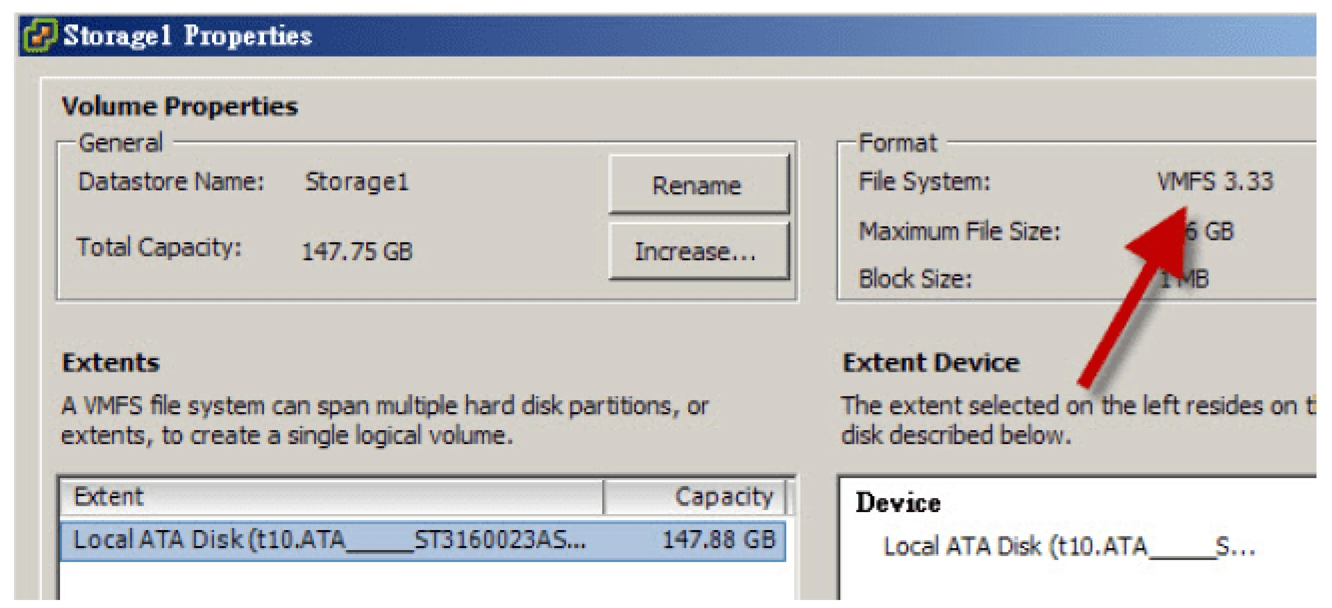
一般来说就是一个存储设备的统称。通常一个本地的逻辑分区、一个 iSCSI 的 LUN、FC SAN 的 LUN 或是一个 NFS 的分享文件夹都可以成为 ESX/ESXi 的 Datastore。在 ESX/ESXi 上，所有的数据都放在 Datastore 中，包括 VMDK 格式的虚拟机和 ISO 文件。在 ESX/ESXi 中，不同的 Datastore 是可以交换数据的。



▲ 在 ESX/ESXi 中，存储的基本单位就是 Datastore

2. VMFS

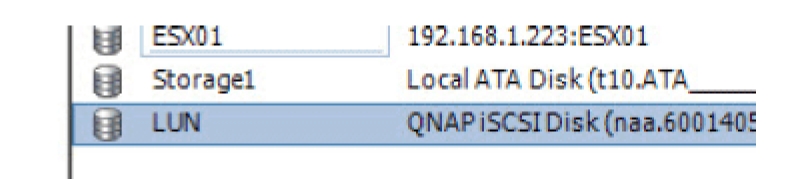
在 ESX/ESXi 之下，除了 NFS 之外，如果是以 LUN 方式呈现的 Datastore 都会被格式化成 VMFS 格式。



▲ VMFS 是所有 Datastore 的文件格式

3. LUN

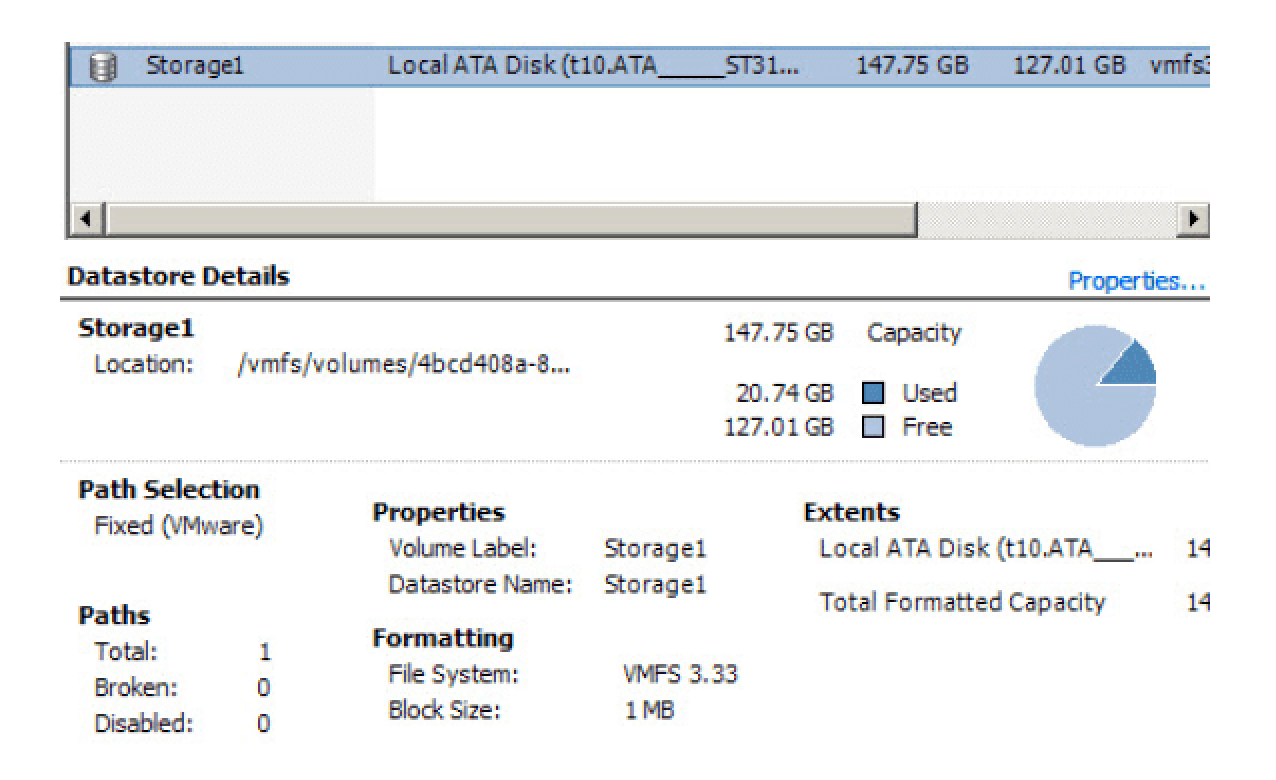
如果是外部存储设备的 LUN，会以 LUN 的方式被给定。LUN 在 ESX/ESXi 下可以被给定成 Datastore，也可以完全直接对应上 RDM，是最有弹性的一种存储设备。



▲ 也可以使用 LUN

11.1.3 内部存储

在 ESX/ESXi 下，置放 VM 的空间都称为 Datastore，不管是本地或是独立设备，在真正开始安装 VM 之前都必须创建 Datastore。在安装 ESX/ESXi 时会默认创建一个。



▲ 安装 ESX/ESXi 会自动创建 Storage1

提示	在第 8 章安装 ESX/ESXi 的部分，我们演示了使用 RAID 0 和 RAID5。大部分使用 ESX/ESXi 初学者，会以为是在安装 Linux，因此在进行 ESX/ESXi 安装的硬盘分区时，也会以为整个硬盘都被分配给 Linux 了。事实上，ESX/ESXi 的 Linux 界面只占了系统很小的空间，因为那只是用来管理 ESX/ESXi，并没有完整的 Linux 功能，大部分的空间，则是分配到另一个分割区，这个分割区就是在本地置放 VM 的具体位置。

1. 本地 Linux 的空间

管理 ESX/ESXi 的 Linux 需要一些空间，称为本地管理分割区。在安装完 ESX/ESXi 之后，我们可以使用传统的 Linux 命令来检视。举例来说，如果我们用一个 160G 的硬盘安装了 ESX/ESXi，在安装完毕之后，在 Linux 的命令行下键入 df –h，可以看到这个 Linux 只占了 8G 的空间，意味着剩下的 152G 的磁盘空间，都已经分配给本地的 VM 来使用了，这个 152G 在 Linux 是看不出来的。

```
root@esxasus:/
[root@esxasus /]#
[root@esxasus /]# df -h
Filesystem            Size  Used Avail Use% Mounted on
/dev/sdc5             4.9G  1.6G  3.0G  35% /
/dev/sda1             1.1G   75M  954M   8% /boot
/dev/sdc2             2.0G   50M  1.8G   3% /var/log
[root@esxasus /]#
```

▲ 本地管理的 Linux 使用的空间极少

2. 置放 VM 的本地存储

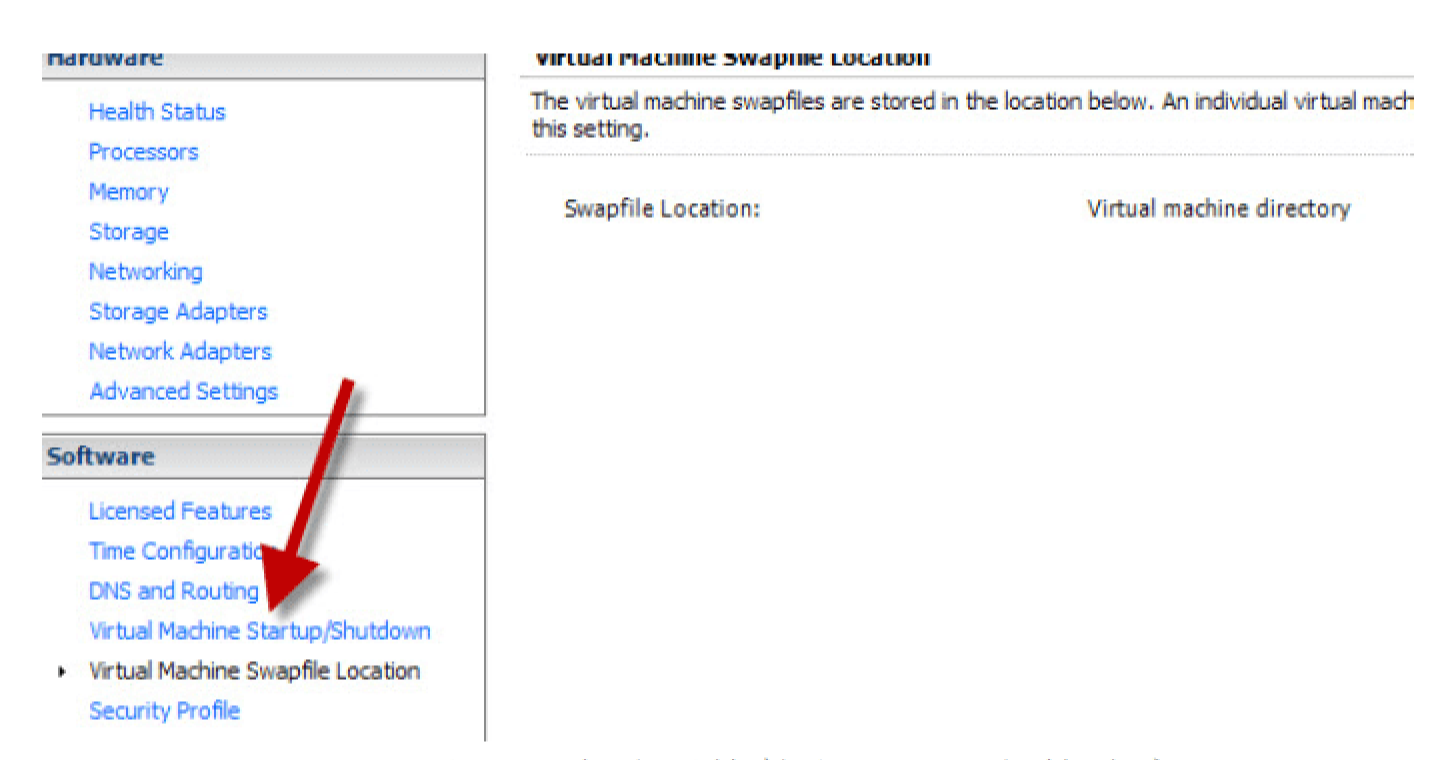
如果在安装 ESX/ESXi 没有改名的话，那么在 ESX/ESXi 下的第一个 Storage 名称就称为 Storage1。这是一个独立的分割区，其格式为 VMFS，为 VMware 独特的文件系统，我们在 Linux 的主控台下也可用 ls 来检视，键入 ls /vmfs/volumes/，就可以看到本地的第一个 Datastore。我们也可以使用 Linux 命令 du –h 来检视这个目录的使用。只要进入该目录，键入 cd /vmfs/volumes/Storage1，并且在其下键入 du –h，即可看到当前这个 Storage1 的磁盘使用状况。

```
root@esxasus:/vmfs/volumes/Storage1
[root@esxasus /]#
[root@esxasus /]# cd /vmfs/volumes/Storage1/
[root@esxasus Storage1]#
[root@esxasus Storage1]# du -h
64K     ./esxconsole-4bcd385c-9e49-0c81-f145-0007e9a56f20/core-dumps
1.4M    ./esxconsole-4bcd385c-9e49-0c81-f145-0007e9a56f20/logs
7.5G    ./esxconsole-4bcd385c-9e49-0c81-f145-0007e9a56f20
2.1G    ./2003
760M    ./ISO
11G     ./c2003
21G     .
[root@esxasus Storage1]#
```

▲ 可以看到本地的存储

3. 置换档空间

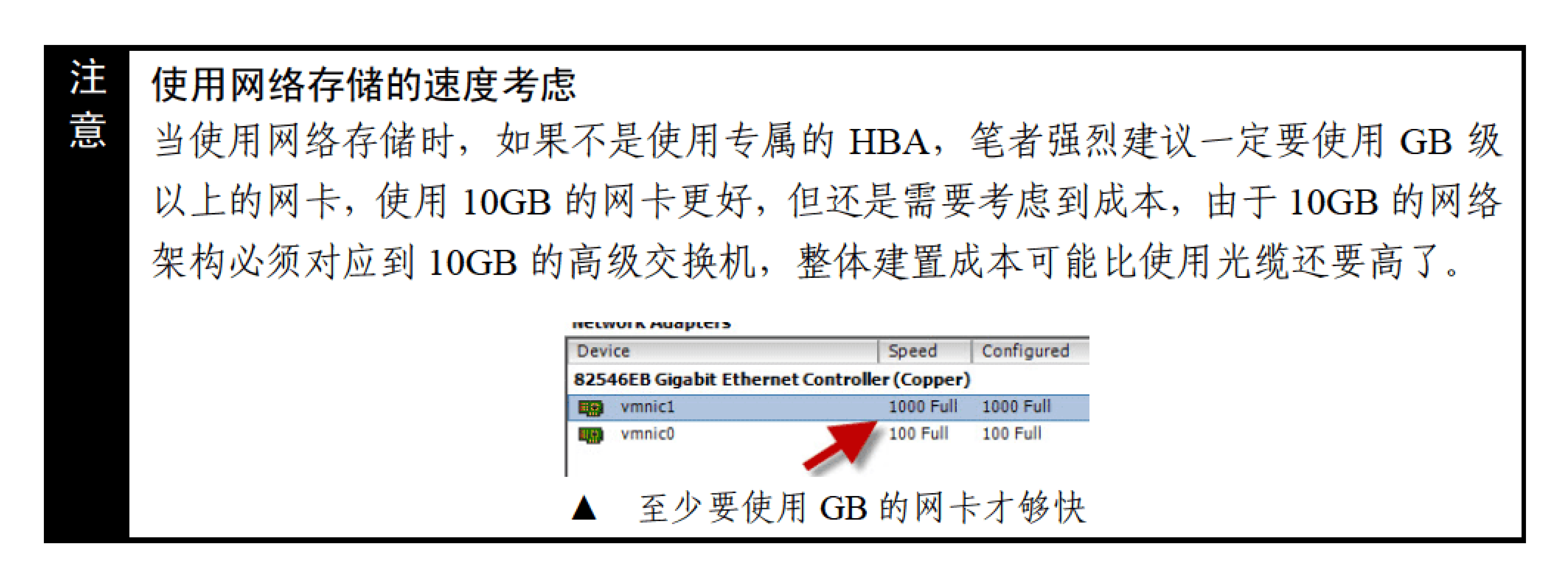
ESX/ESXi 本身的 Linux 以及 Hypervisor 也像一般的操作系统需要置换文件，因此有一个 Swap 分割区。但要特别注意的是，这个置换档空间是专门给 Linux/Hypervisor 用的，VM 中的操作系统所使用的置换文件则需要特别配置位置，在稍后的章节有详细说明。



▲ 配置 VM 中的置换档空间可以加快速度

11.1.4 ESX/ESXi 的外部存储

ESX/ESXi 可以连接 DAS、NAS 以及 FC SAN，连接到 ESX/ESXi 上之后，也必须以 Datastore 的方式存在，不过 ESX/ESXi 也支持 RDM 模式，可以直接可读外部存储的单个逻辑磁盘（LUN）。除了 RDM 之外，ESX/ESXi 最常见的外部存储为 NFS、HBA 直接存储及 iSCSI 三种。



注意

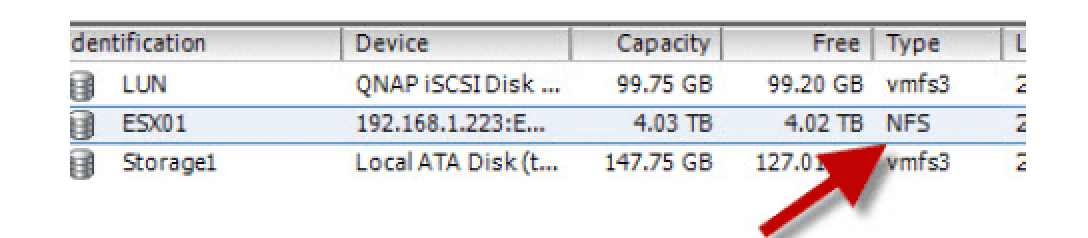
使用网络存储的速度考虑

当使用网络存储时，如果不是使用专属的 HBA，笔者强烈建议一定要使用 GB 级以上的网卡，使用 10GB 的网卡更好，但还是需要考虑到成本，由于 10GB 的网络架构必须对应到 10GB 的高级交换机，整体建置成本可能比使用光缆还要高了。

▲ 至少要使用 GB 的网卡才够快

1. NFS

NFS 是 Linux 的磁盘共享机制，只要在 Linux 磁盘或是支持 NFS 的独立存储配置好账号口令，就可以在 ESX/ESXi 中加入 Datastore 了。



▲ NFS 是使用 EXT4 来置放文件

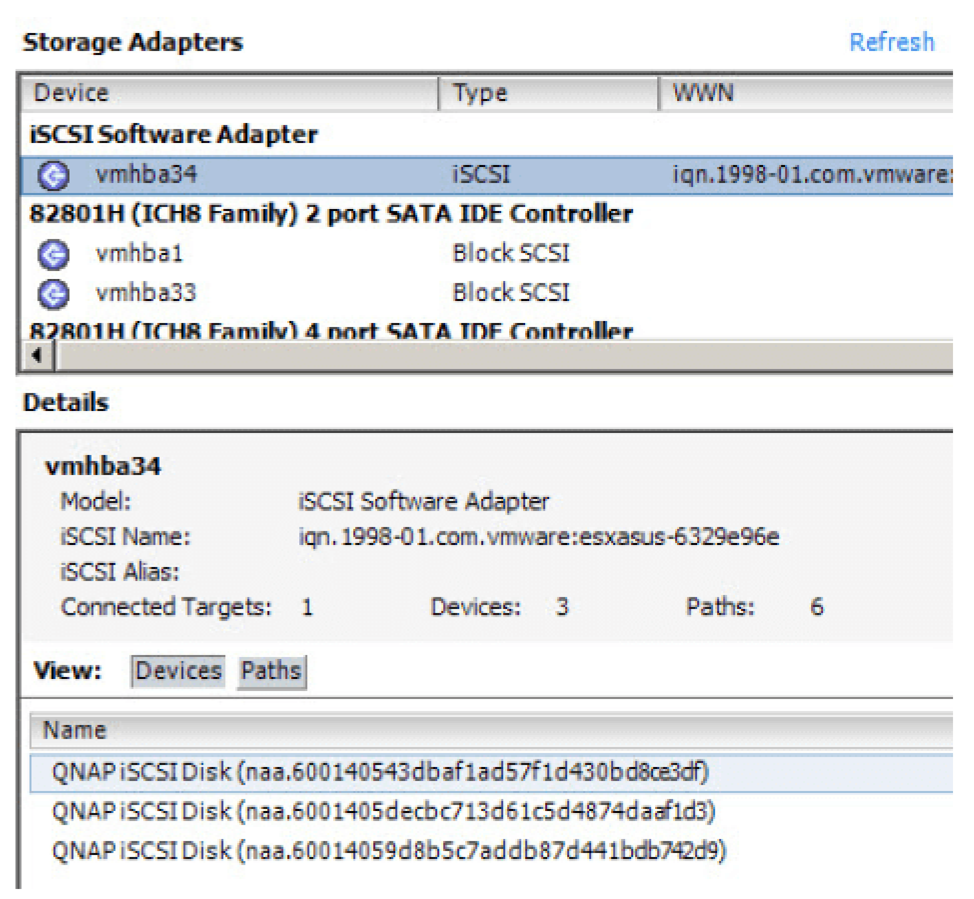
2. HBA 直接存储

如果你的 ESX/ESXi 有安装专属的 HBA 卡，就可以用 LUN 或是专属的方式连上外部存储，专业的 HBA 卡包括了 iSCSI、FC SAN 等，使用专业的 HBA 卡主要原因就是传输的速度，快捷的 HBA 卡可以到 5GB 的速度，并且不占主系统的 CPU 资源，但成本也较高。在本书高级存储应用的章节，我们会介绍 FC SAN 的访问方式。

3. iSCSI 连接

iSCSI 是使用网卡当做 HBA 卡的一种机制，当然也有专属的 iSCSI HBA，但在成本及性能

的平衡下，使用 GB 网卡作为 iSCSI 的连接是最常见的外部存储连接方式。本书大部分的存储示例都会以 iSCSI 作为演示。



▲ 在 ESX/ESXi 下 iSCSI 的弹性较大

11.2 ESX/ESXi 上的外部存储：NFS

NFS（Network File System）是最常见的网络磁盘共享机制，是属于 ESX/ESXi 下 NAS 连接的一种。NFS 必须先创建自身的文件系统，大部分为 EXT4，并且创建账号口令。在 ESX/ESXi 下，连接到 NFS 的方式主要为网络，我们就来看看在 ESX/ESXi 如何连接上 NFS 的外部存储。

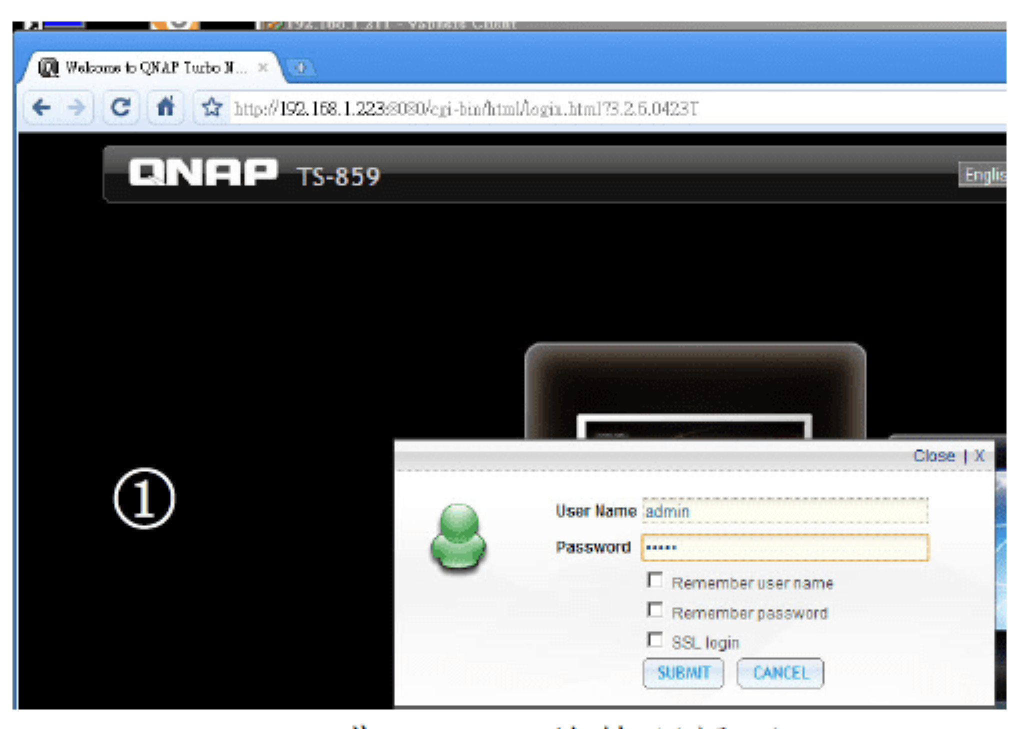
11.2.1 准备 NFS 共享磁盘

要在 vSphere 下使用 NFS，首先就是要拥有一个提供 NFS 的设备。你可以自行安装 Linux，并且将其中的文件分享出来，但最好用的还是专业的 NAS，我们就以专业的 NAS 作为示例。

1. 创建 NAS 上的 NFS 共享

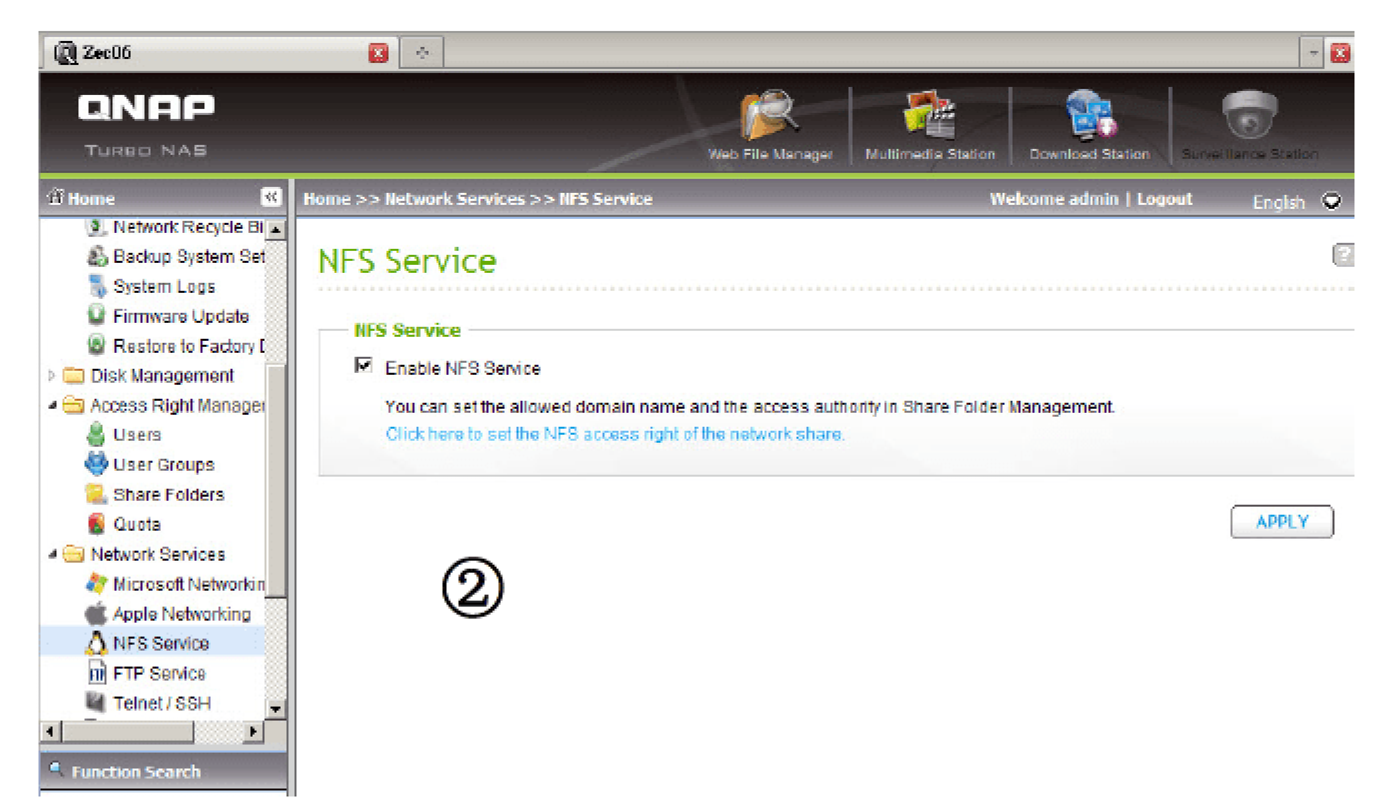
NFS 共享是以 IP 为主，而分享的基本文件是文件夹，因此我们必须先将共享文件夹创建，并且配置访问的账号口令，就可以进行访问了。下面就是详细步骤。

1. 本例以 QNAP 的 TS-859 NAS 作为演示。首先进入其配置界面，键入账号口令。



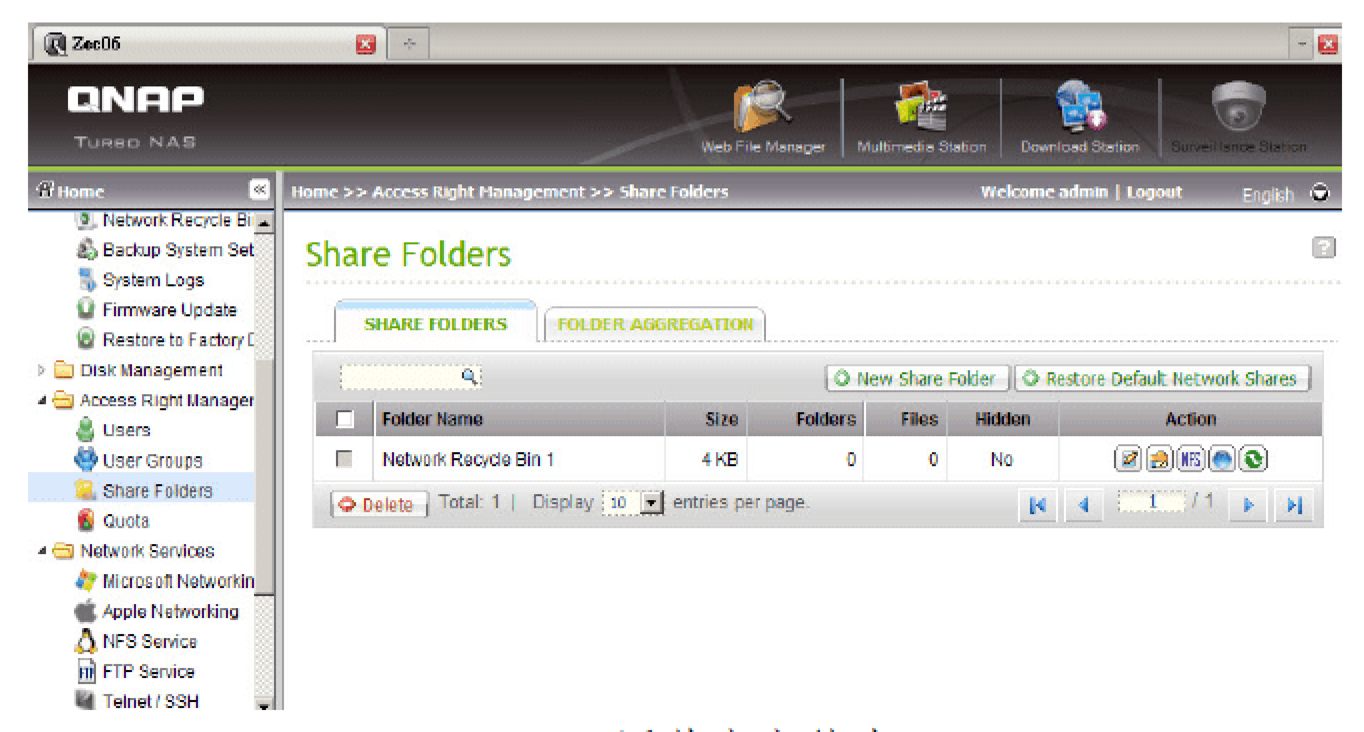
▲ 进入 NAS 的管理界面

2. 选择 Network Services NFS Service 选项。将 Enable NFS Service 打勾，并且单击 Apply 按钮。



▲ 先激活 NFS 功能

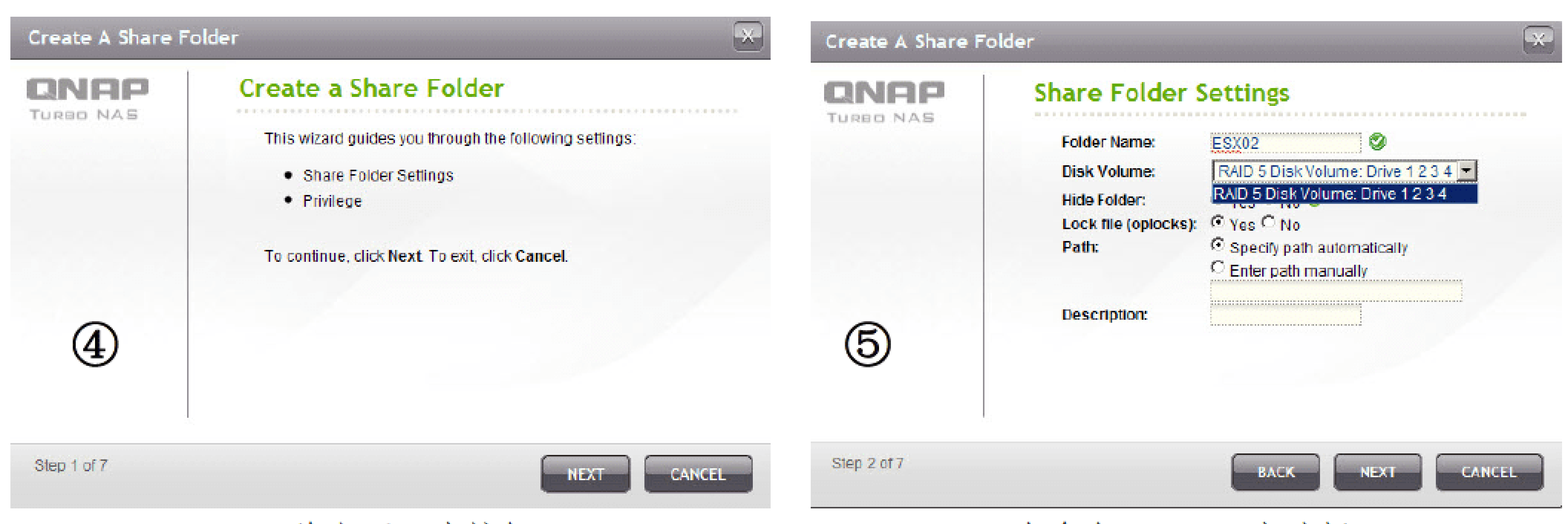
3. 在上图中，我们激活了 Click here to set the NFS access right 的连接。此时会进入 NFS 的配置画面。首先选择 New Share Folder 选项卡。



▲ 配置共享文件夹

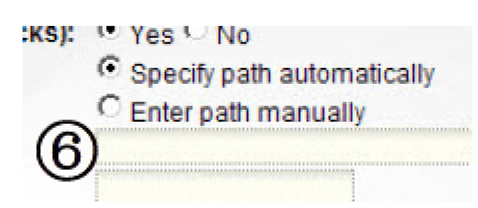
4. 弹出如下图所示的画面时，单击 Next 按钮。

5. 首先我们创建一个文件夹的名称。接着在 NAS 上选择一个已创建好的 Volume，如下图所示的 RAID 5。



▲ 单击下一步按钮　　　　▲ 在多个 Volume 中选择

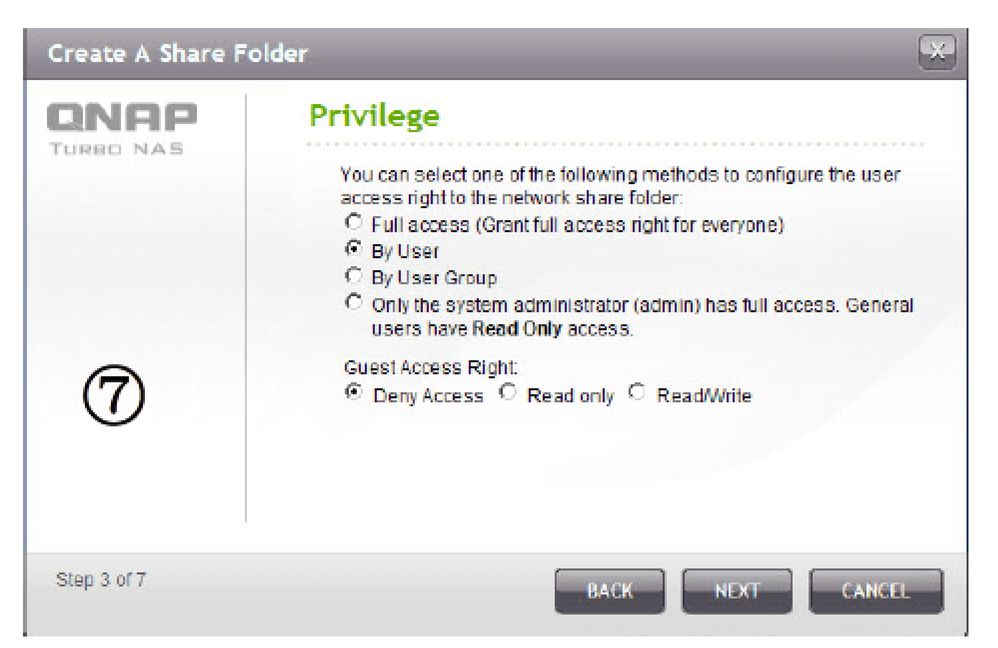
6. 我们可以将 Specify path automatically 打勾，落实之后单击 Next 按钮。



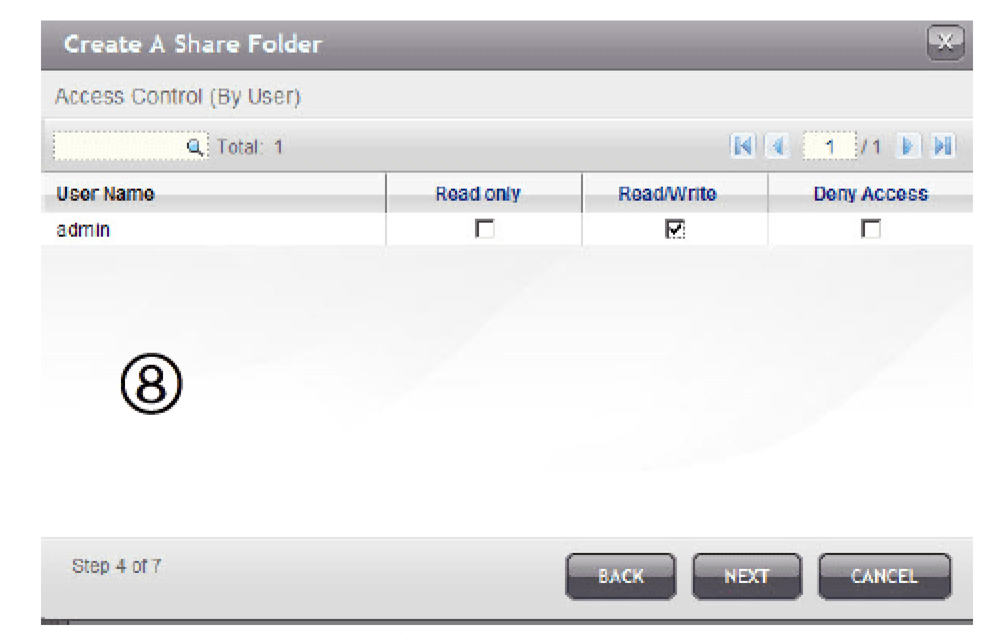
▲ 选择这个地方，在 ESX 上不需要键入目录即可

7. 由于我们要访问 NFS 需要可写的权利，因此选取 By User 单选按钮。之后单击 Next 按钮。

8. 接下来弹出这台 NAS 的用户，我们将 Read/Write 选中，单击 Next 按钮。



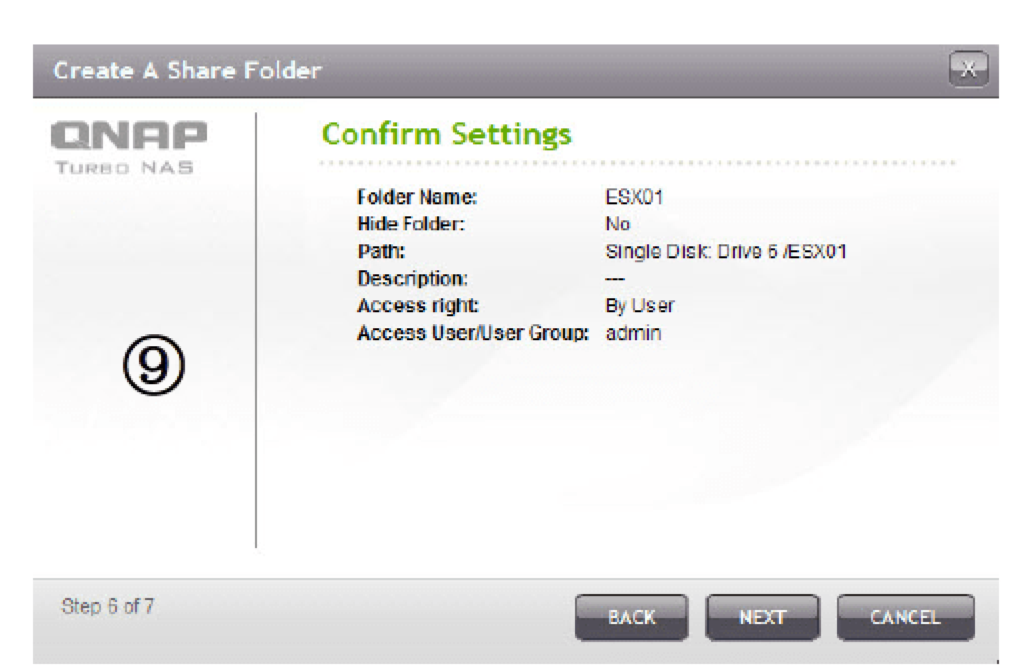
▲ 这里先配置 By user，之后还要开放



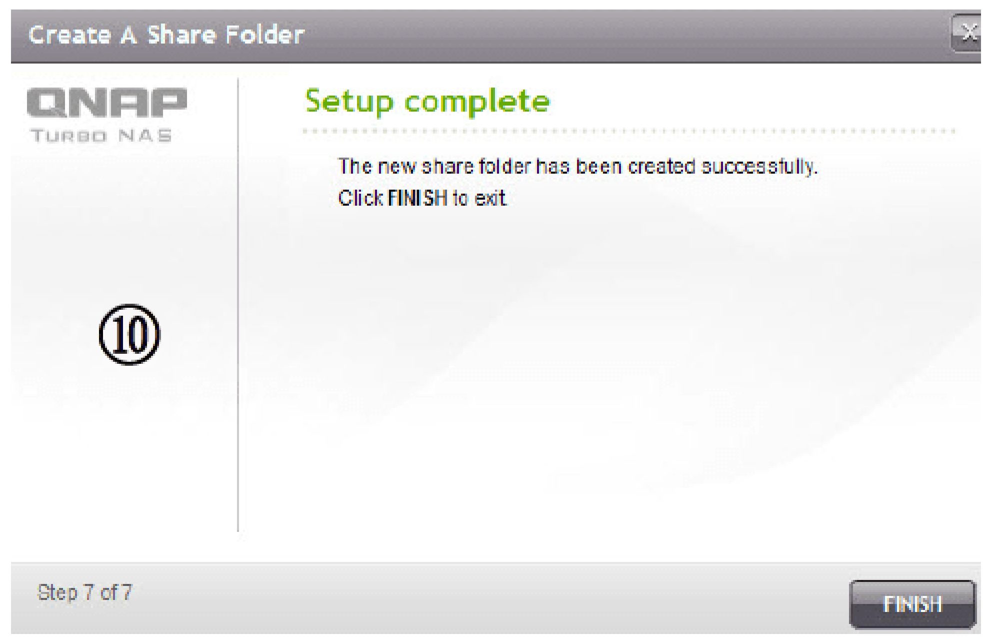
▲ 读写权利都要有

9. 接下来会弹出总结画面，单击 Next 按钮。

10. 此时会弹出创建的画面。单击 FINISH 按钮即可。

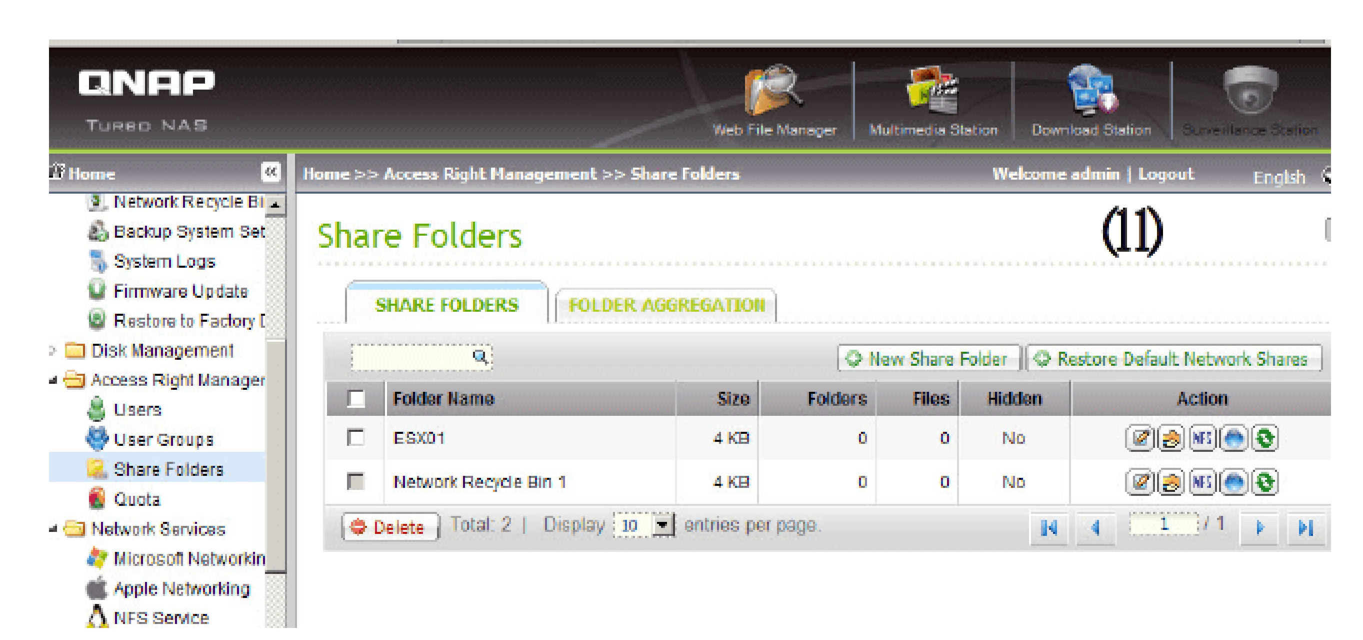


▲ 总结画面



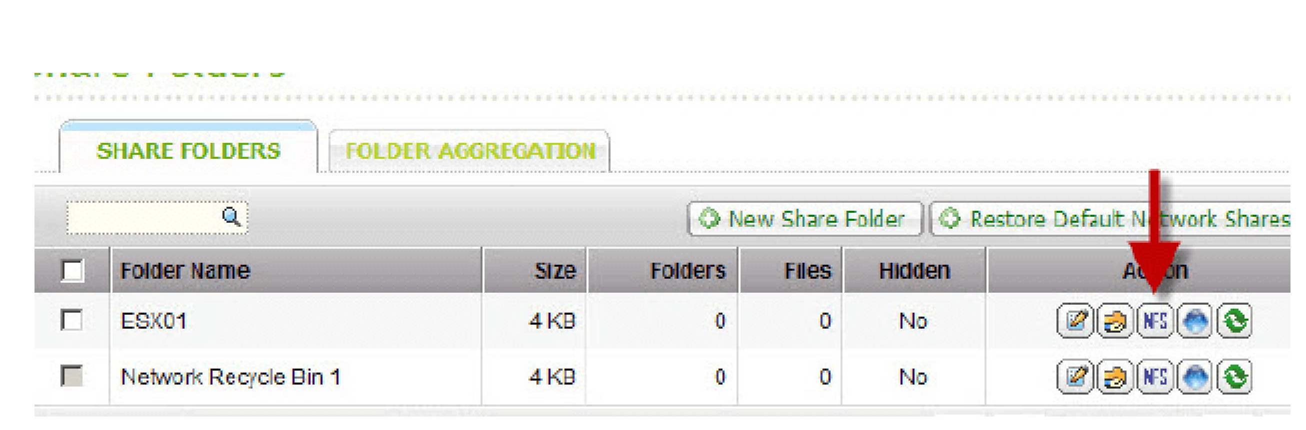
▲ 创建完毕

11. 我们会看到在图中弹出一个 ESX/ESXi01 的文件夹，这就是我们要分享的 NFS。单击 NFS 按钮。

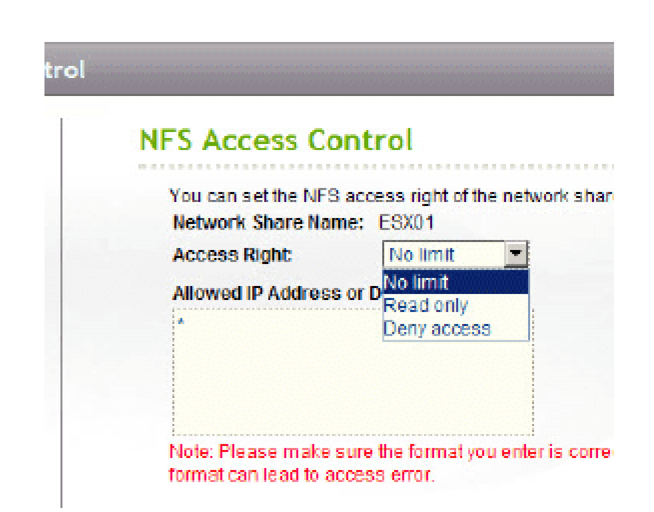


▲ 这就是创建好的 NFS

12. 进入之后，在 Access Right 中选择 No limit 选项。单击 Apply 按钮。



▲ 从这里进入

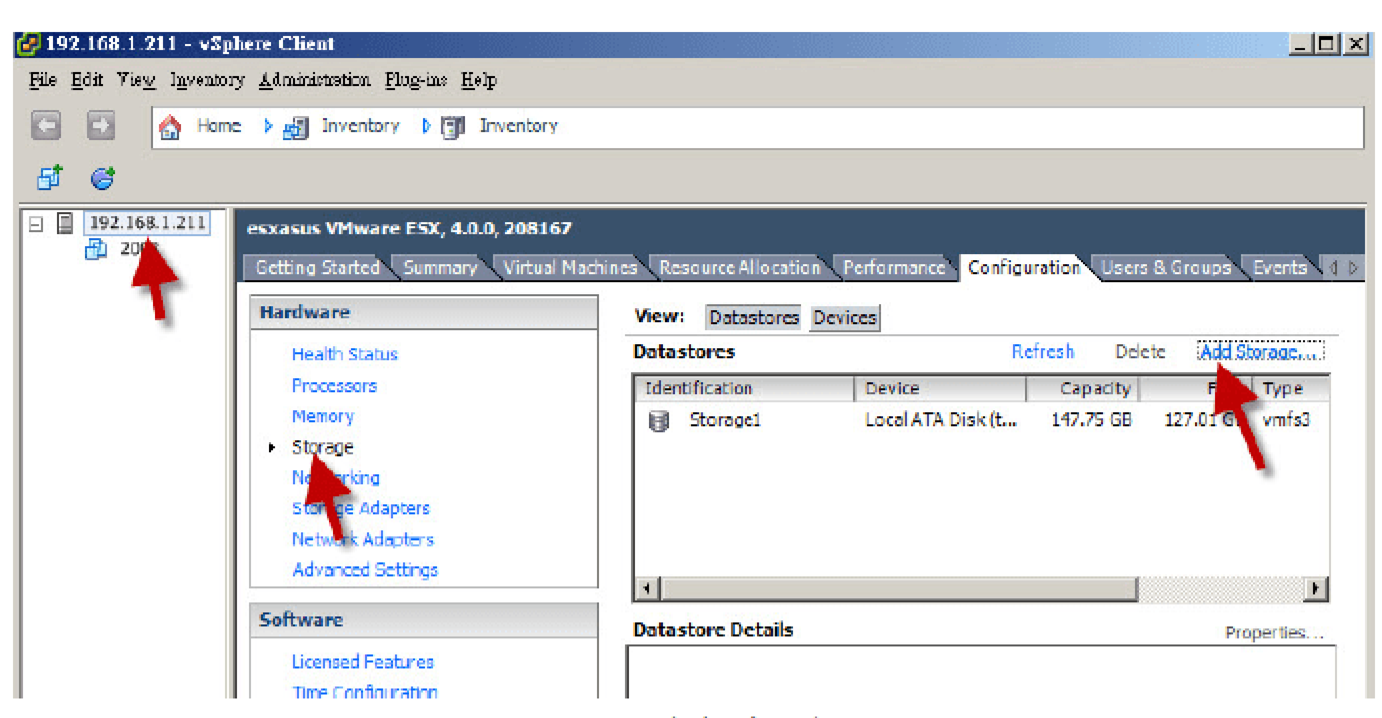


▲ 这里要不制约才能让 ESX/ESXi 连入

2. 在 ESX/ESXi 上添加 NFS 为 Datastore

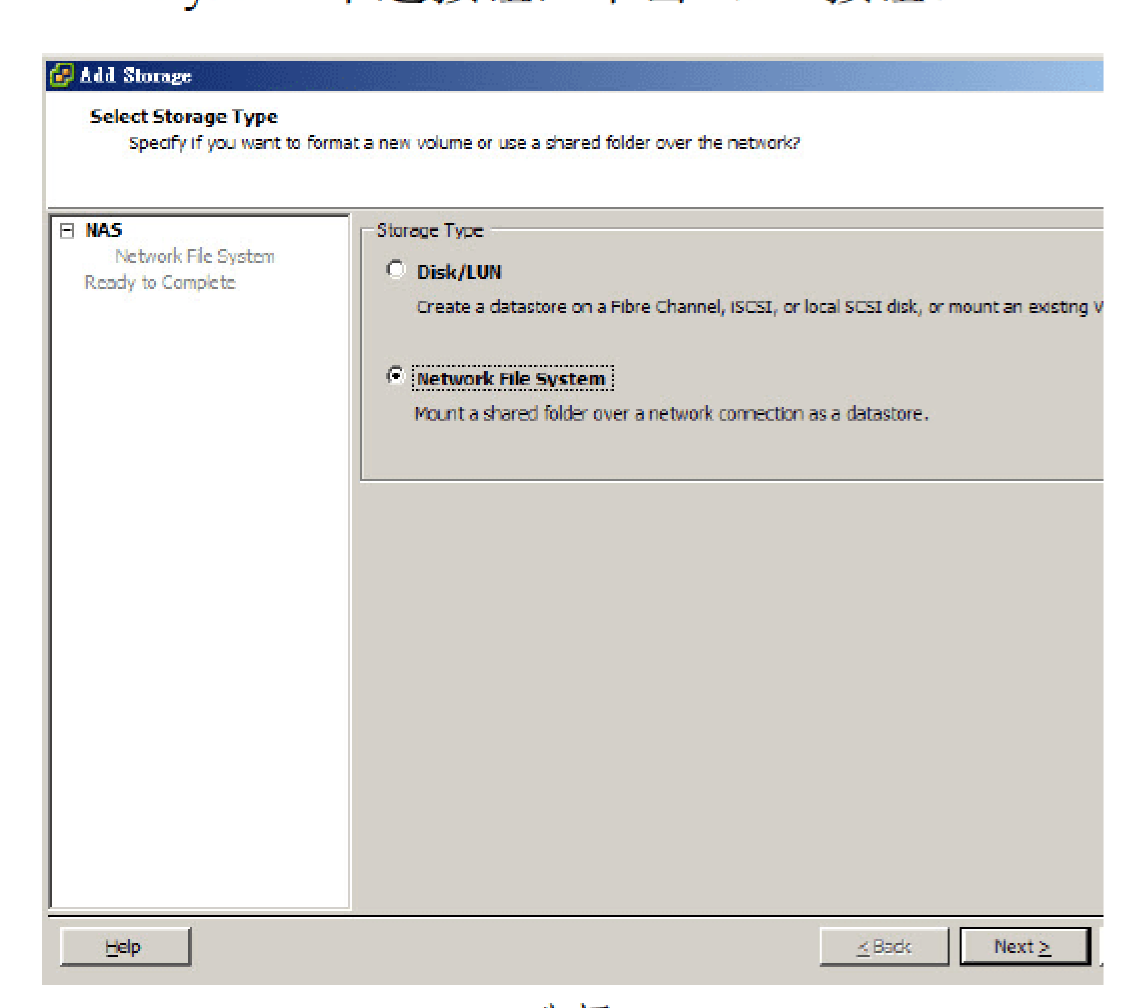
在 ESX/ESXi 上配置 NFS 的步骤就是加入存储设备的步骤。

1. 进入 vSphere Client，并且选择你要加入 NFS 的 ESX/ESXi 服务器。选择 Configureation/Storage/Add Storage 选项。



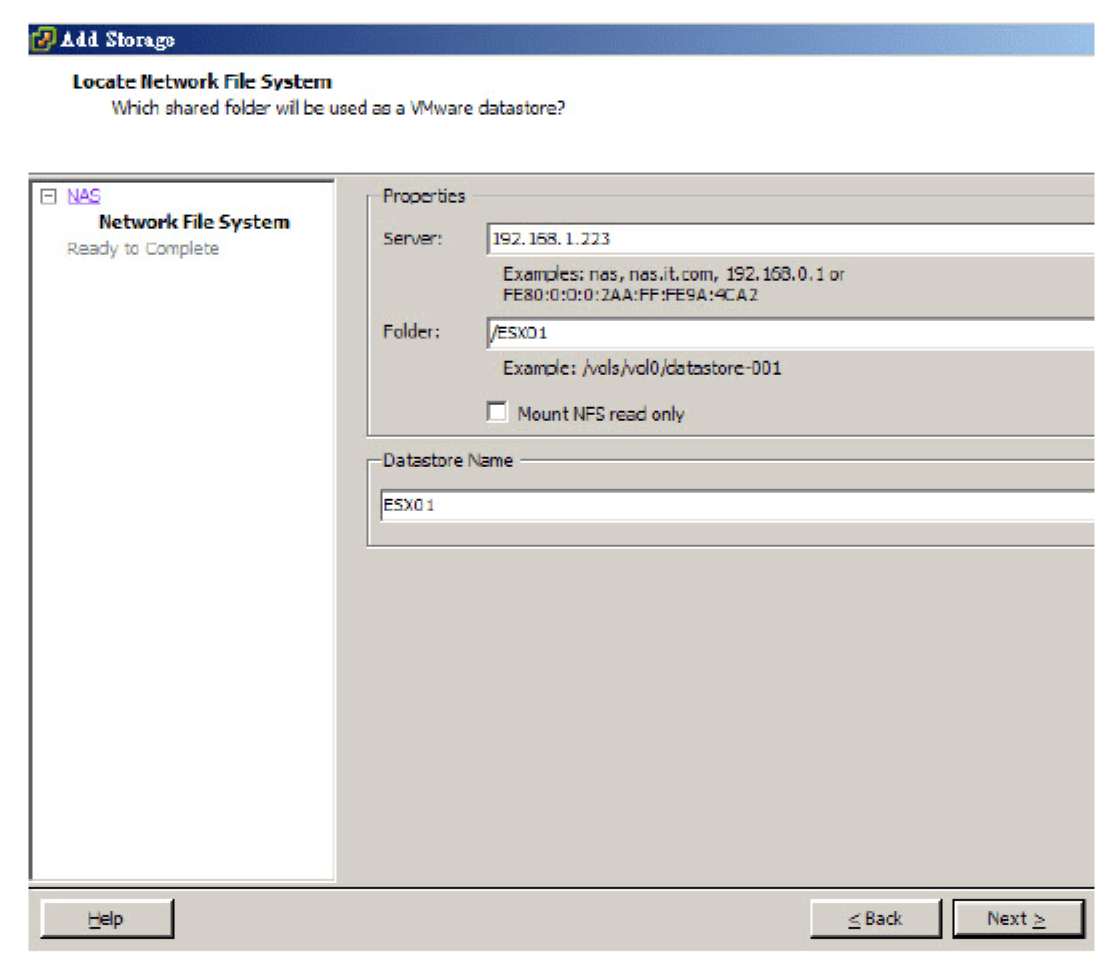
▲ 选择新建

2. 选择 Network File System 单选按钮，单击 Next 按钮。



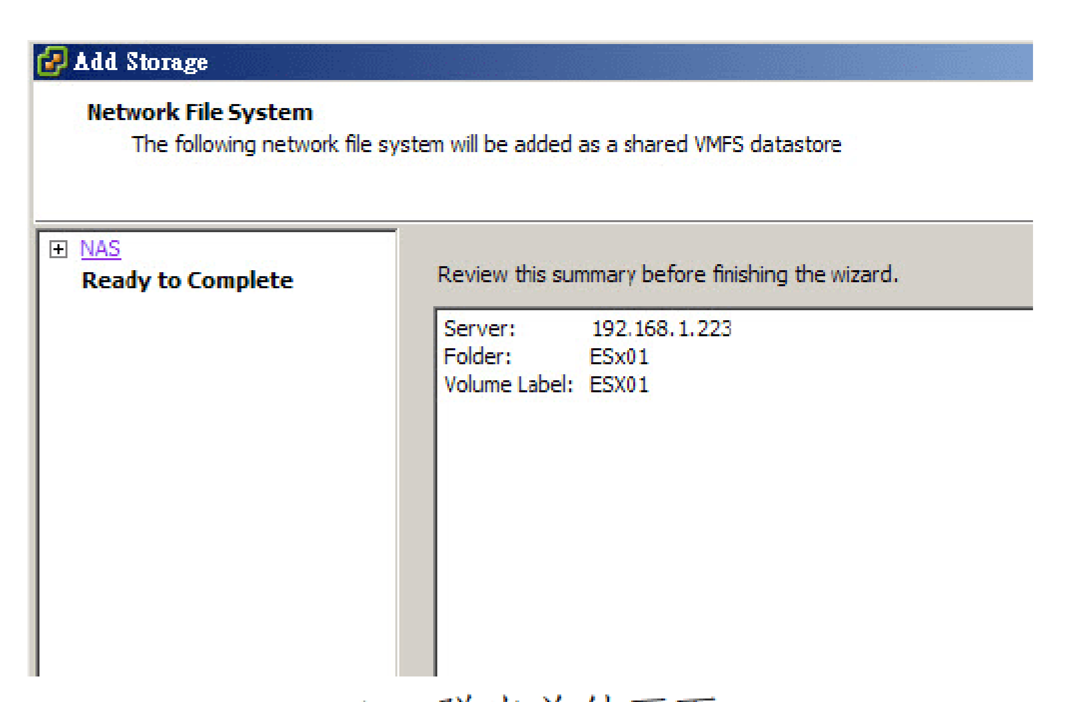
▲ 选择 NFS

3. 在接下来的窗口，键入 NAS 的 IP，以及文件夹名称。我们并发也可以给定在 ESX/ESXi 下 Datastore 的名称。键入完之后单击 Next 按钮。



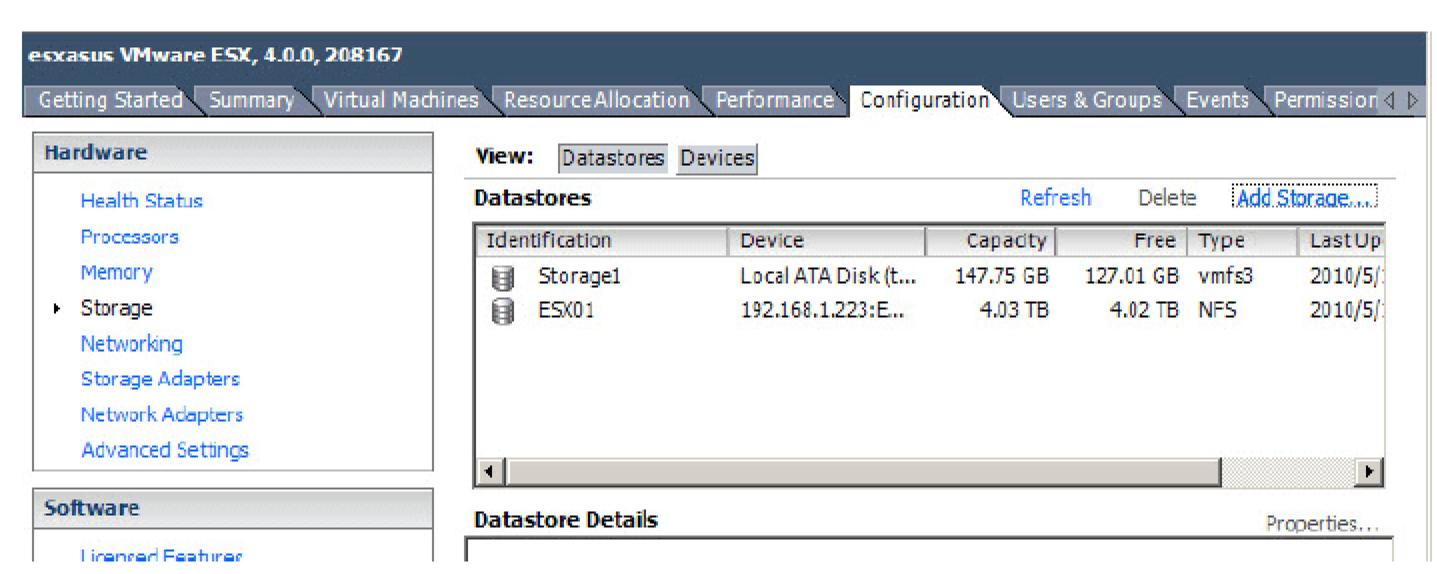
▲ 键入 IP、共享名称以及 Datastore 的名称

4．之后会弹出总结画面，单击 FINISH 按钮即可。



▲ 弹出总结画面

5．之后我们就可以在 ESX/ESXi 的 Storage 上看到这个 NFS 的 Datastore。



▲ 在 ESX/ESXi 下也弹出了

11.3 ESX/ESXi 上的外部存储：iSCSI

我们在安装 ESX/ESXi 时，使用了本地的硬盘，因此在安装虚拟机时也将其安装在 ESX/ESXi 的本地硬盘中。但这么做的坏处就是当这台 ESX/ESXi 蓝屏时，上面的虚拟机也跟着蓝屏了。在 vSphere 架构中，最有名的 HA、DRS 和 VMotion 这些功能就是为了防止这种灾难的发生，而我们在本书的开宗名义也说到，要落实这些高级的功能一定要使用独立的存储。现在既然 ESX/ESXi

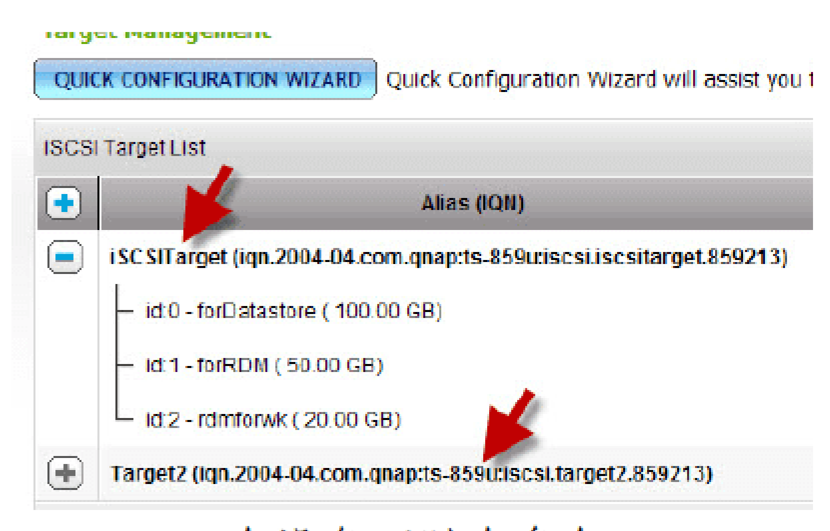
已经安装完毕，接下来我们只要将 ESX/ESXi 连上 iSCSI 的 Target Server 就大功告成了。

11.3.1 准备 iSCSI Target Server

准备一个专门的 iSCSI 存储设备有许多种选择，包括 FC SAN 上面也可以将 LUN 配置成 iSCSI，当前较专业的 NAS 也可以配置 iSCSI 的 LUN 给 ESX/ESXi 使用，我们在这里，还是使用 QNAP 的 TS-859 产品来配置 iSCSI 的 LUN 给 ESX/ESXi 使用。

1. 理解 iSCSI 中的 Target

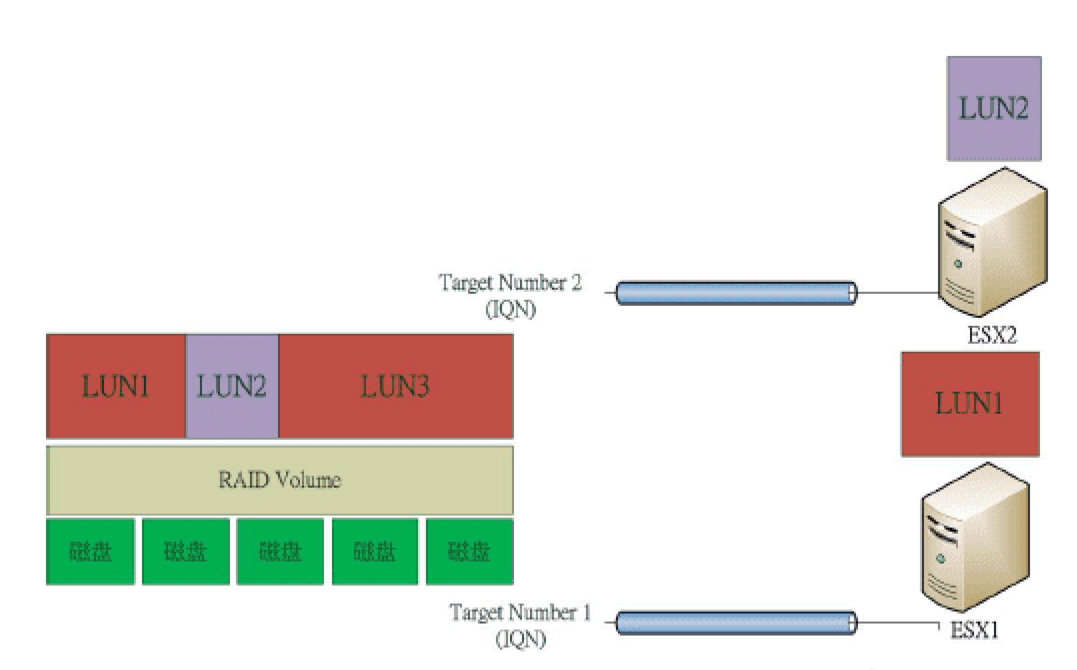
一个 iSCSI 的 Target 就是一个设备提供联机的名称，这会由系统自动给定，而这个 Target 上也可以提供多个不同的 LUN 给外部使用。当前大部分的 NAS 都有提供多 iSCSI Target 的功能，这是为了让多个网段能访问同一个 NAS 设备的目的。在进行 Target 配置时，系统会生成一个全世界唯一的 Target Number，称为 IQN（iSCSI Qualified Name），这个类似互联网域名的号码，是用来辨识 Target 的手段。由于 iSCSI 是通过 IP 提交的，因此有可能跨地区，需要一个唯一的号码。



▲ 一个设备可以有多个 Target

2. 理解每个 Target 中的 LUN

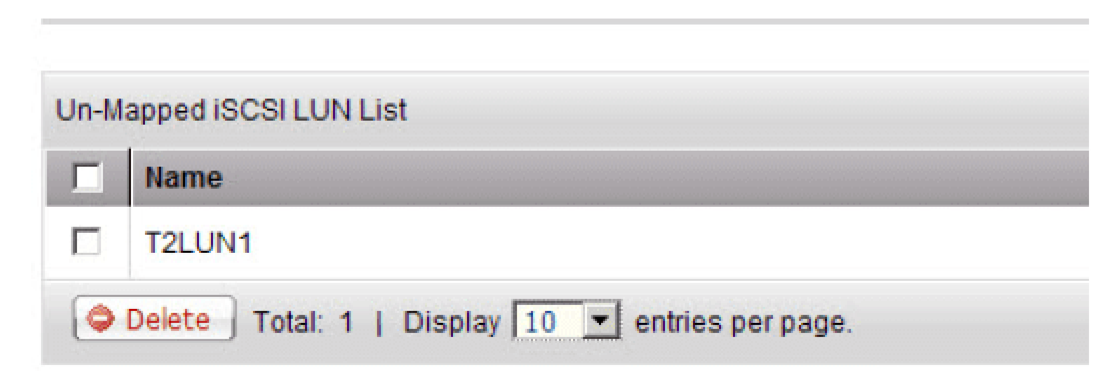
每一个 Target 可以提供多个 LUN，并且让不同的系统可以 Mapping 上。所谓的 LUN 就是不同的分割区。你可以把一台 iSCSI 的存储设备想象成一个硬盘组，每一个 Target 想象成一个“界面卡”，而 LUN Mapping 就是这台大硬盘上不同的分割区，用分割区为单位来挂载上 ESX/ESXi，不同的网段通过不同的“界面卡”来访问这个大硬盘，因而达成空间分享的目的，可引用下图所示的说明。



▲ 图为设备、Target 和 LUN 的关系

3. LUN 的 Mapping

每个 Target 下的 LUN，并不一定要在创建好之后立马创建联机使用。要联机时，需要创建 Mapping，就是将 LUN“映射”出去。你可以将这个 Mapping 的动作想象成“上线”的概念，可创建很多个 LUN，但真正需要时才将它们 Mapping 出去。



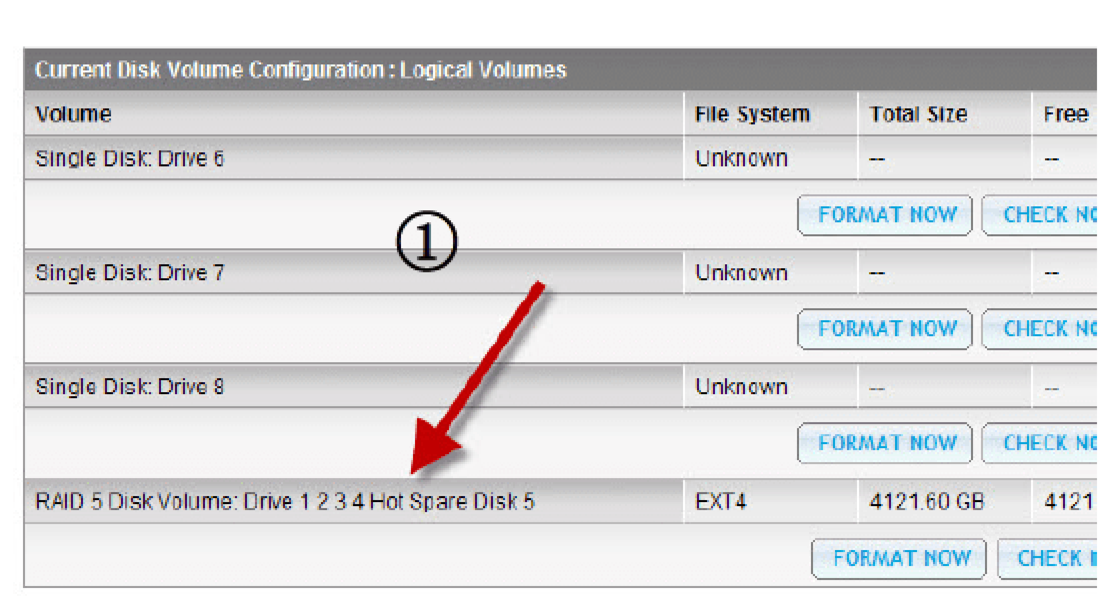
▲ 创建 Target 后不一定要 Mapping

4. 创建 QNAP 的 iSCSI Target Server

一个 iSCSI Target Server 主要是提供存储空间和网络连接。以 LUN 作为单位进行连接。因此在连接之后，ESX/ESXi 会看到一个完整的逻辑磁盘，并且在其上创建 VMFS 文件系统而非 NAS 提供的文件系统。

▶ 创建 iSCSI 的 Target Server

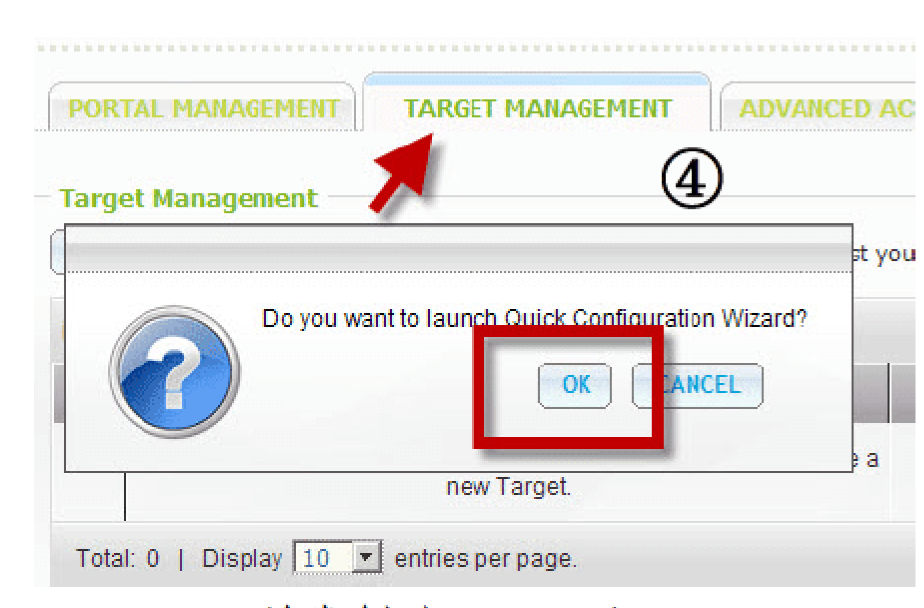
1. 首先确定我们的 NAS 上已经创建好 Volumes 了。一般 NAS 的 iSCSI LUN 是存在于已经创建好的 Volumes 上。
2. 进入 QNAP 的 NAS 配置，选择 Disk Management 中的 iSCSI 选项。



▲ 要先创建好 Volumes ▲ 选择这里

3. 在 Portal Management 选项卡中，先将 iSCSI 的服务激活，并且记下通信端口的号码，图为 3260，也是标准 iSCSI 的通信端口，之后单击 Apply 按钮。
4. 接下来要将这台 NAS 配置成一个 iSCSI 的 Target。选择 Target Management 选项系统会弹出是否要并发配置 LUN Mapping。单击 OK 按钮。



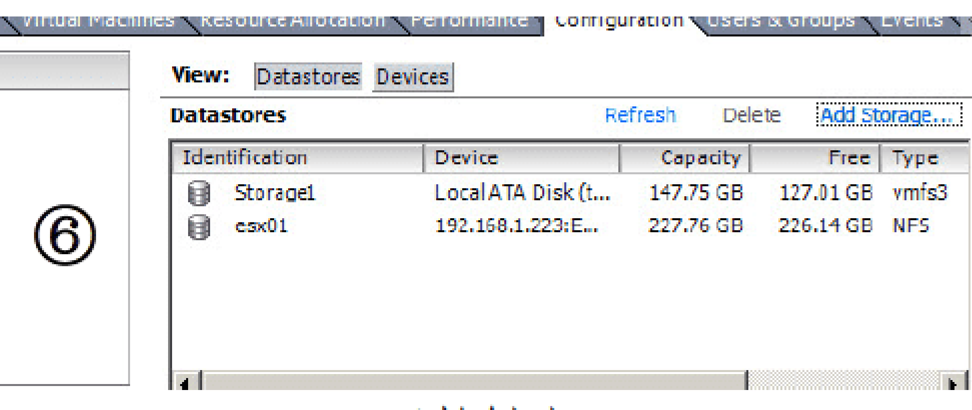


▲ 要先激活 iSCSI ▲ 并发创建 Target 和 LUN

5. 在接下来弹出的窗口中，选择第一个选项，创建 iSCSI 的目标连接，并发创建 LUN，单击 NEXT 按钮。

6．在接下来的如图所示的窗口中单击 NEXT 按钮。

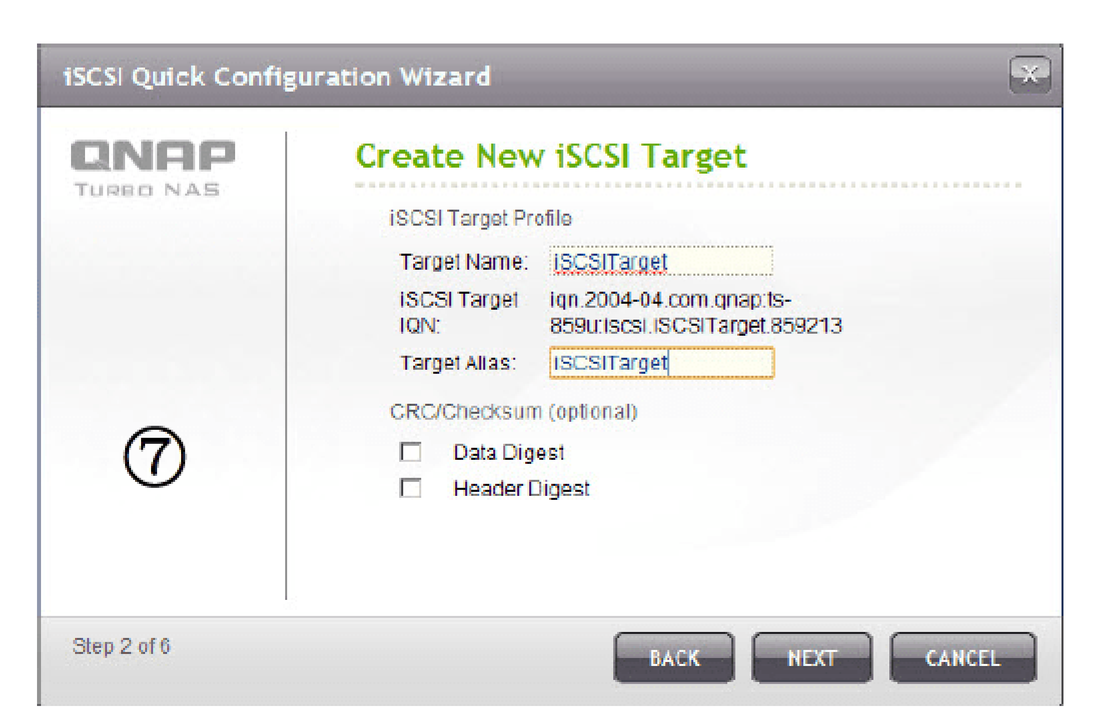




▲ 创建的 LUN 会挂在新建的 Target 下　　▲ 开始创建

7．接下来会要求你键入一个 Target 的名称，我们键入 iSCSITarget。重要的是键入别名，也键入 iSCSITarget。然后单击 NEXT 按钮。

8．接下来是账号口令的键入，你可以在此键入账号口令，从 ESX/ESXi 连接过来时，就需要这个账号口令。键入完之后单击 NEXT 按钮。

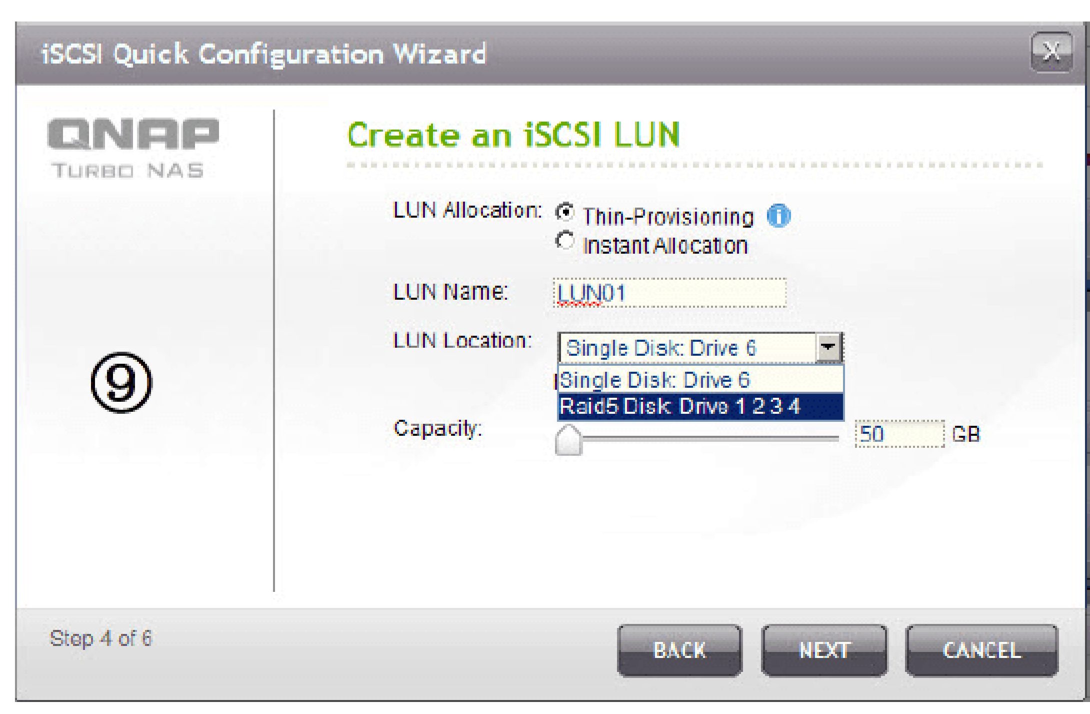


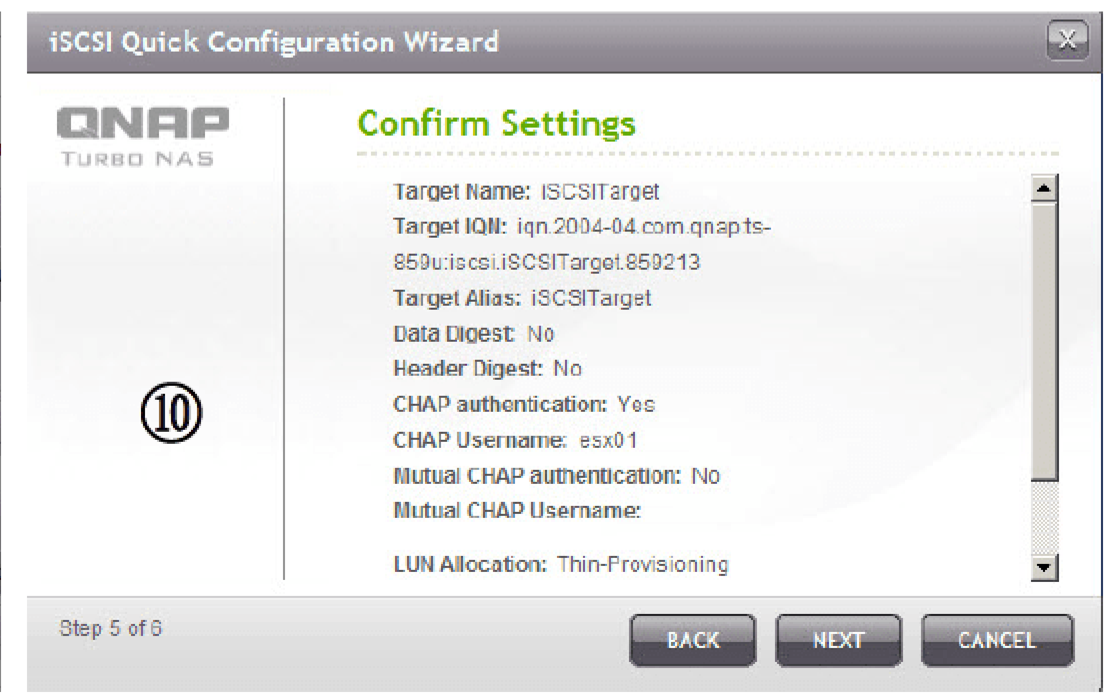


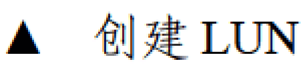
▲ 一般不会用 IQN 连接，因此一定要创建别名　　▲ 用 CHAP 是最常见的认证方式

9．接下来创建 LUN，这里只是帮助你创建第一个 LUN，你也可以在稍后手动创建更多的 LUN。我们选择 Thin Provisioning 单选按钮，会需要时才分配空间，并且键入 LUN 的名称，如 LUN01，接下来从 NAS 中的 Volumes 中选择一个创建的位置，并且在随后给定大小，落实后单击 NEXT 按钮。

10．接下来会弹出总结画面，单击 NEXT 按钮即可。

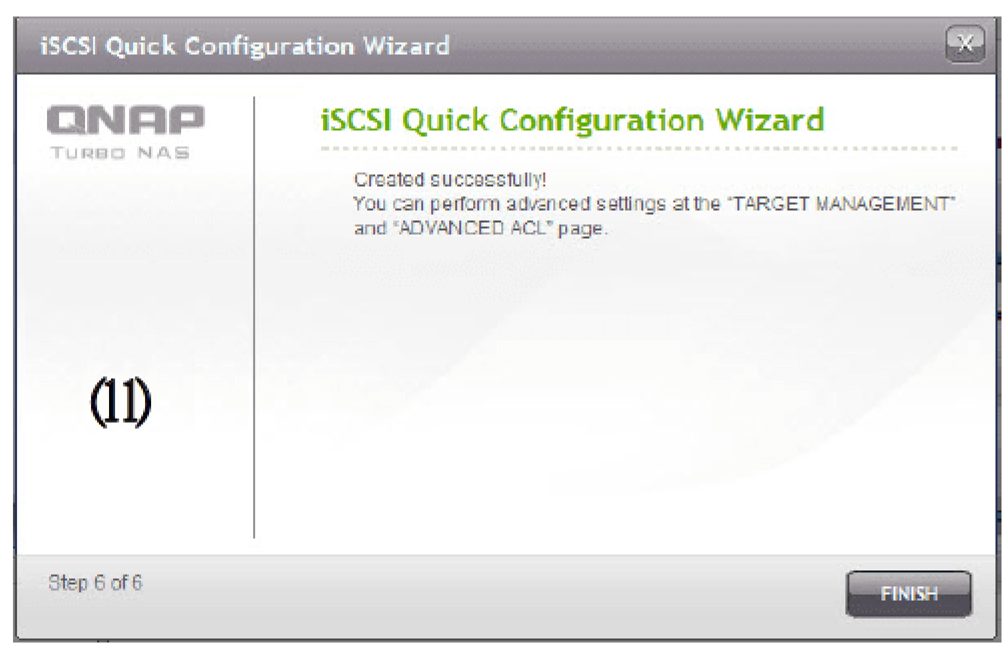


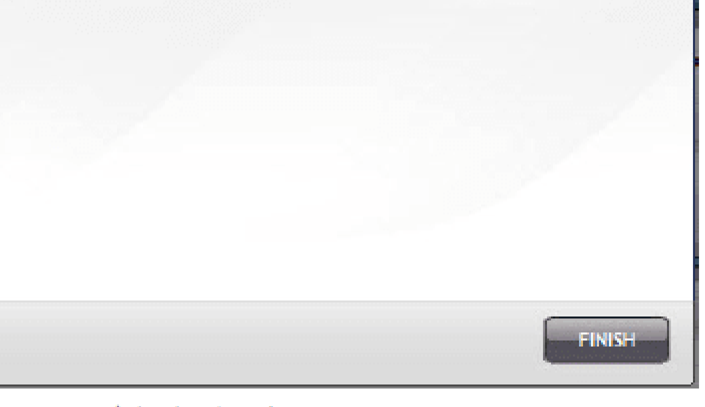



▲ 创建 LUN　　▲ 总结画面

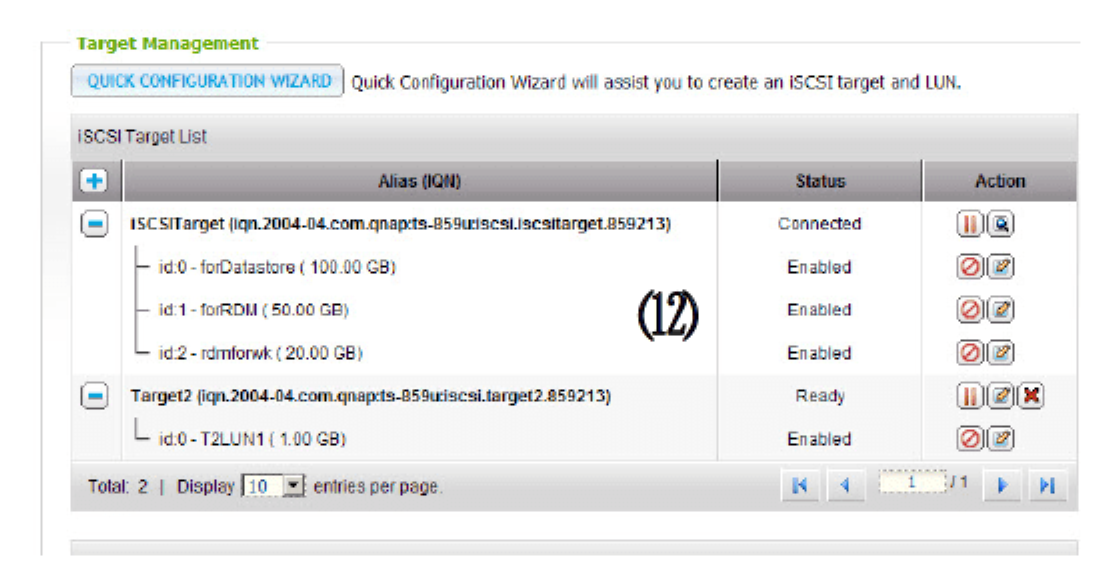
11. 此时系统会开始创建 LUN，当弹出如下图所示的画面时，意味着创建完毕，单击 FINISH 按钮落实。

12. 我们可以从如下图所示的画面中看到这台 NAS 上可创建多个 Target，而每一个 Target 下也有自身的 LUN，并且会显示已经创建，但还没有 Map 的 LUN。





▲ 创建完毕



▲ 图中有 Target 及 LUN

11.3.2 准备 ESX/ESXi 服务器

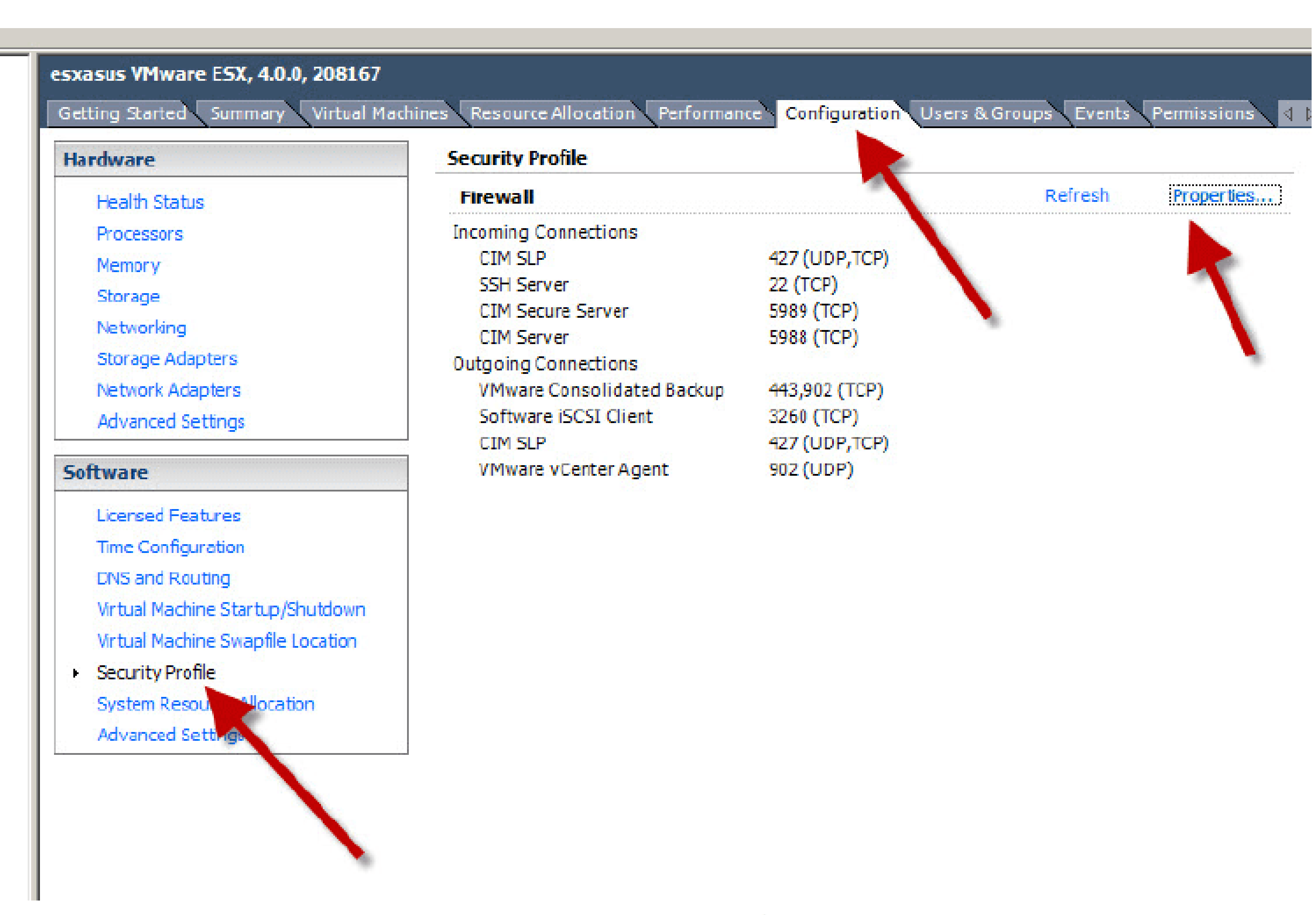
要让 ESX/ESXi 服务器能连上 iSCSI Target Server，必须在 ESX/ESXi 中配置一个 iSCSI 的通信端口，开放防火墙，并且创建 iSCSI 的 HBA 界面。

1. 开放 ESX/ESXi 的防火墙

ESX/ESXi 服务器本身即为一个 Linux 内核激活的 Hypervisor，因此也具有 Linux 的防火墙。我们可以进入主控台中将 iSCSI 的通信打开，但这需要使用 iptable 的命令，相对于 Linux 较不熟悉的用户会比较麻烦。还好 vSphere Client 中已经有这个功能了，可以通过其界面打开 iSCSI 的通信端口。如果读者们使用的是 ESX/ESXii，则不需要打开 iSCSI 的防火墙，可以直接使用。我们就来看看如何打开 ESX/ESXi 中的防火墙。

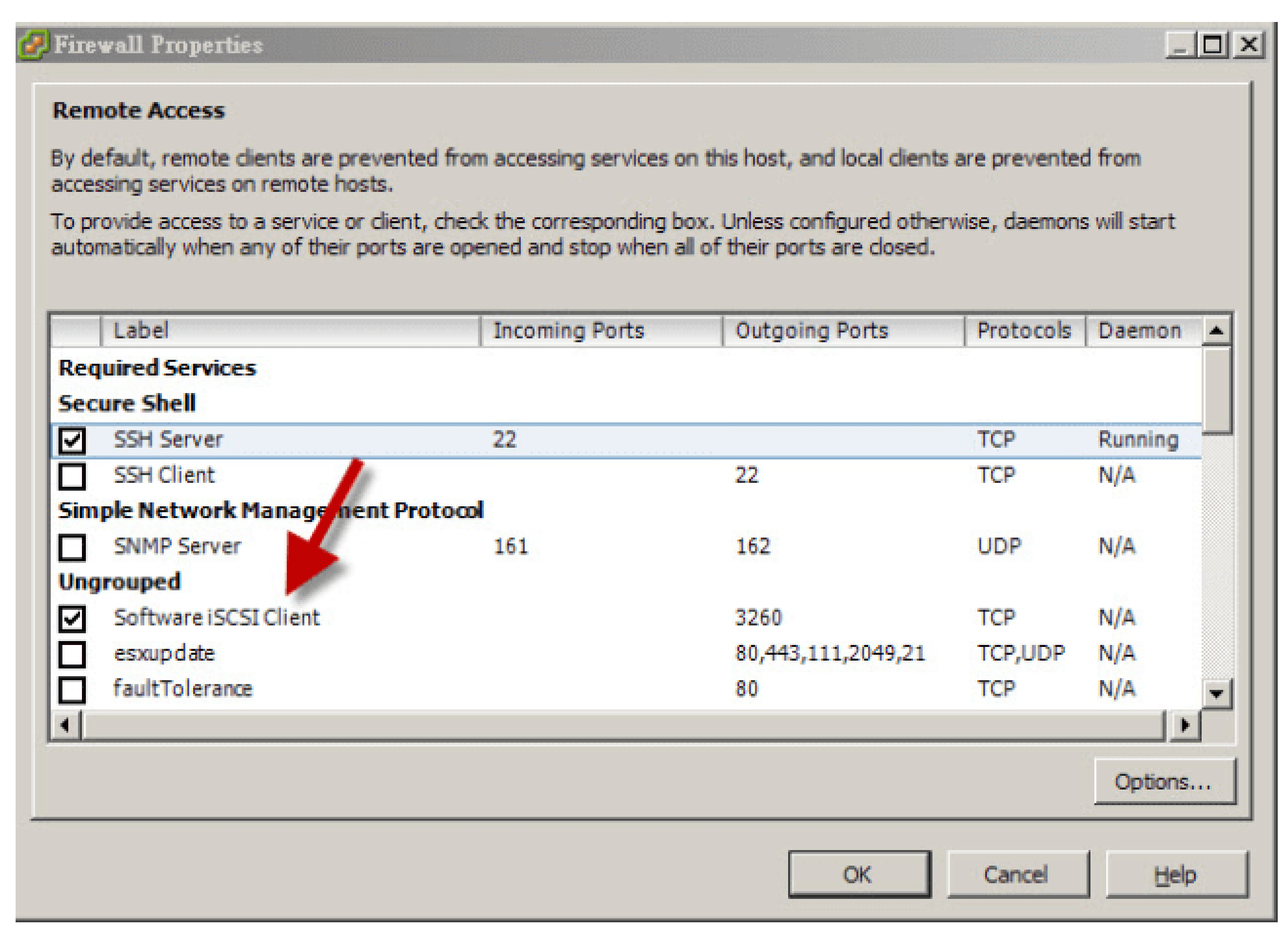
▶ 打开 ESX/ESXi 中的 iSCSI Client 防火墙

1. 进入 vSphere Client 的 Configuration / Security / Profile / Properties。



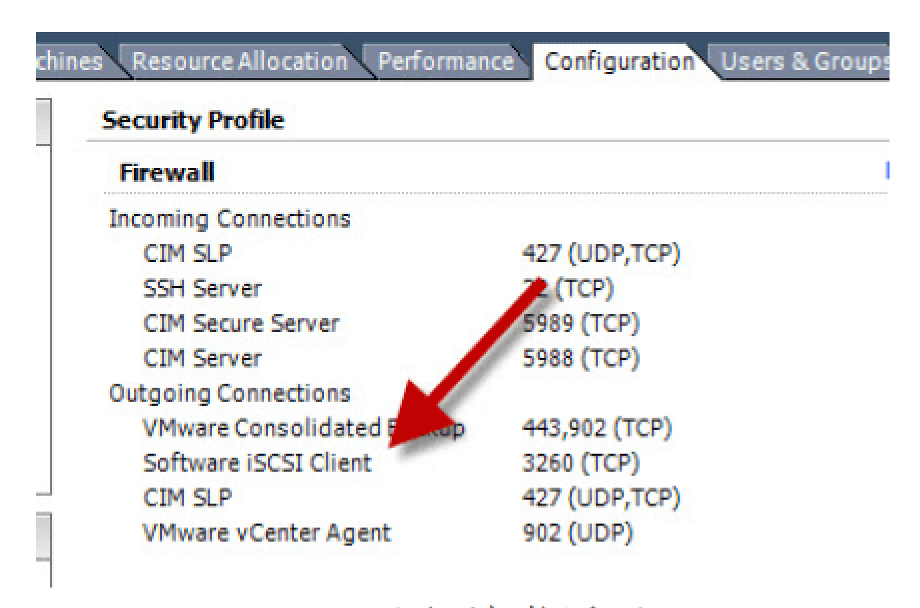
▲ 选择安全策略

2．选择 Software iSCSI Client 复选框。



▲ 选中 iSCSI

3．再回到主画面，我们已经可以看到 iSCSI 已经激活了。

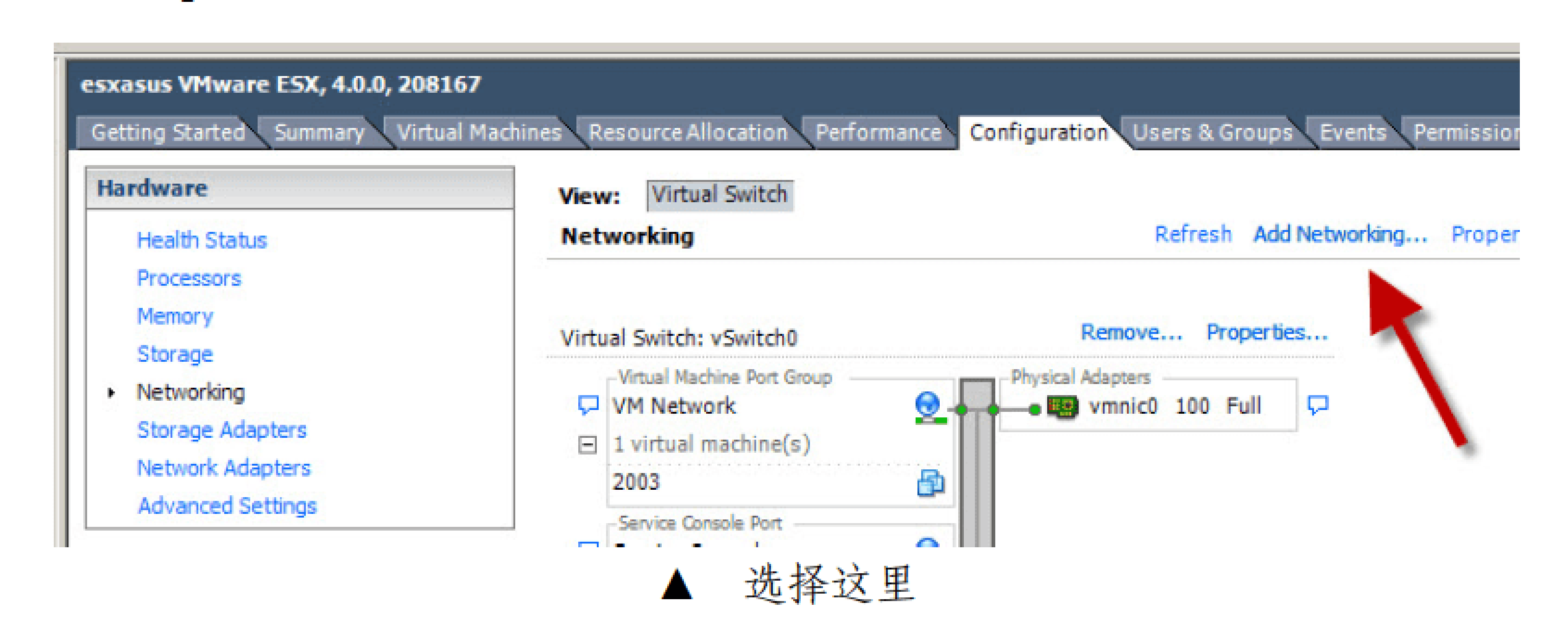


▲ 已经被激活了

2. 创建 iSCSI 虚拟路由器和通信端口

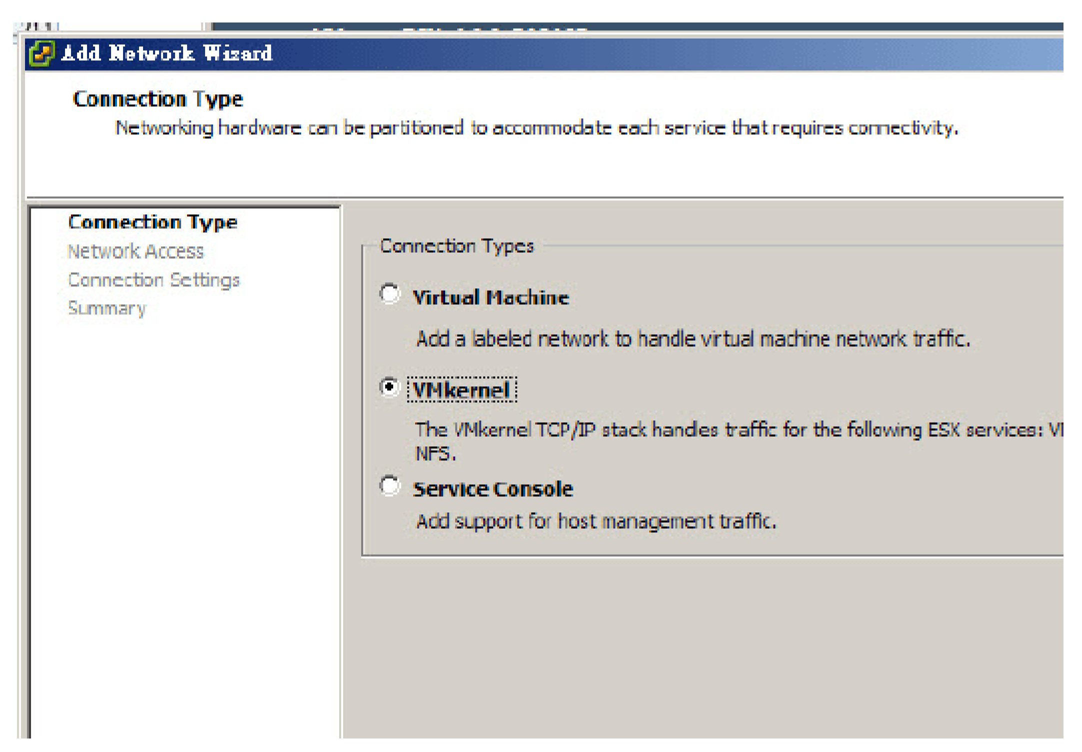
为了系统的稳定，我们可以使用一个单个的网卡，创建一个单个的虚拟交换机，并且配置一个单个的 VMkernel 通信端口给 iSCSI 使用，并且给定一个固化 IP。我们就用这个示例来复习一下 ESX/ESXi 的网络观念。

1．进入 vSphere Client 中的网络配置部分，选择新建网络。



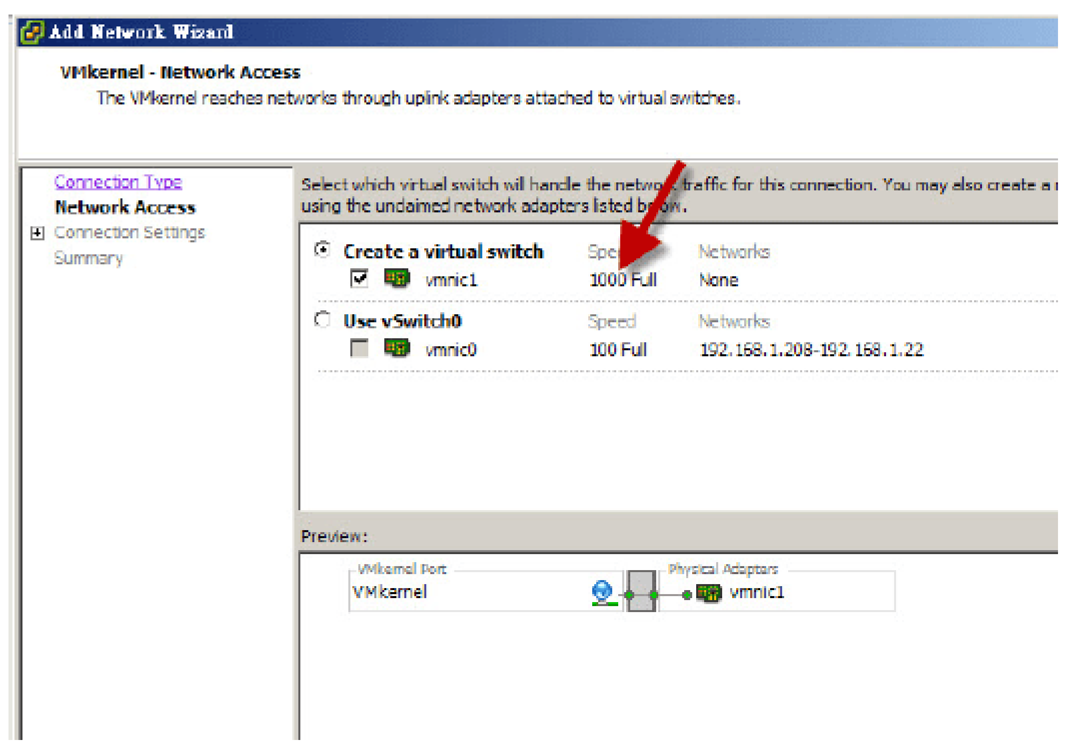
▲ 选择这里

2．选择创建 VMkernel 的通信端口。



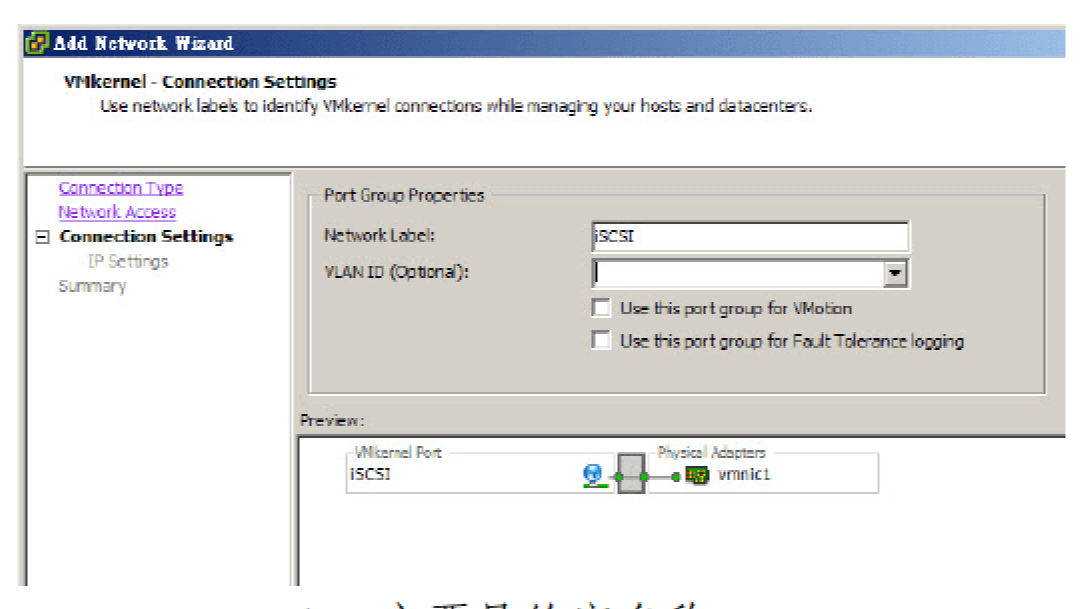
▲ 虽然 ESX/ESXi 系统会默认在主控台的网卡创建 HBA，但也可以自行创建

3．选择创建新的虚拟交换机，并且创建在新的 vmnic 1 上。



▲ 选择第 2 片网卡

4．选择创建 iSCSI 的连接，并且给定 IP。

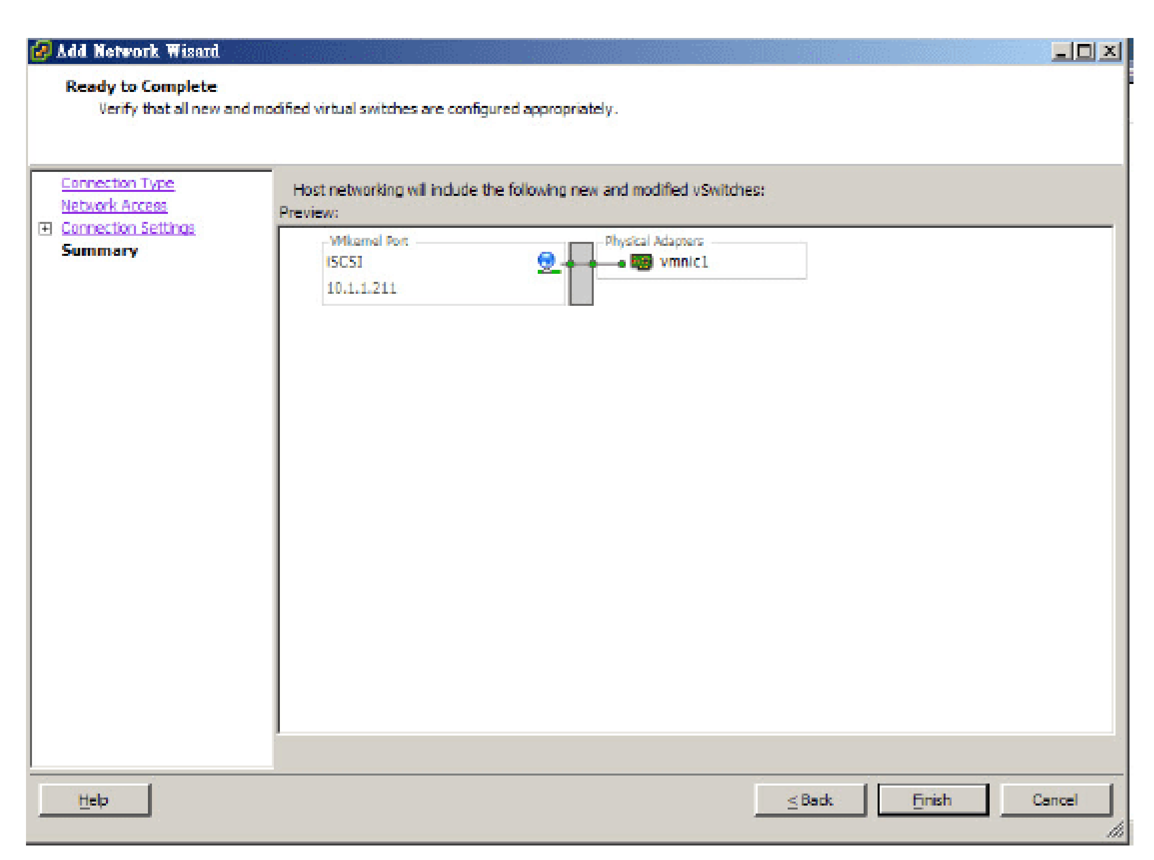


▲ 主要是给定名称

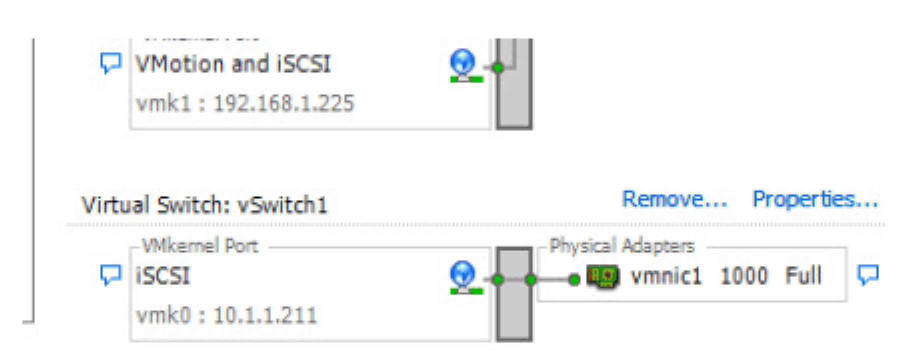
▲ 键入这个连接的 IP

5．之后会弹出结论窗口，单击 Finish 按钮。

6．此时已经看到我们新创建的 iSCSI 界面了。




▲ 新创建的 iSCSI 界面



▲ 创建 iSCSI 完毕

11.3.3 创建 ESX/ESXi 的 Datastore

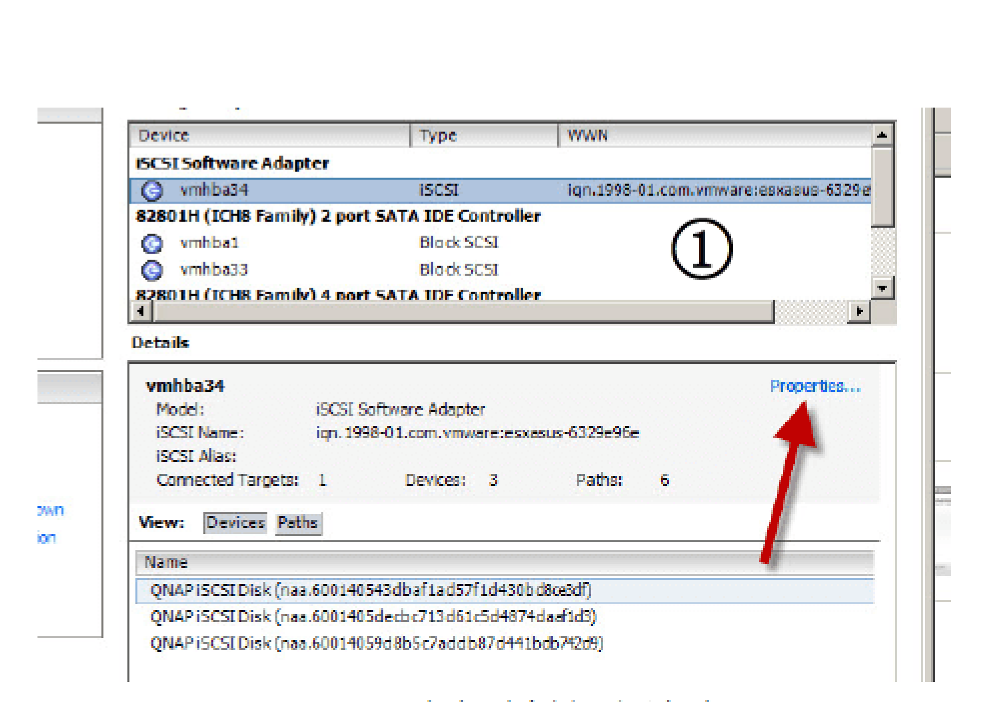
在 ESX/ESXi 和 iSCSI 的准备任务告一段落之后，接下来就是进行 ESX/ESXi 下 Datastore 的创建。我们就以 ESX/ESXi 连接 Openfiler 作为示例，看看在 ESX/ESXi 之下是如何使用软件 iSCSI Initiator 来连入 iSCSI 的 LUN 中。

1. 添加 ESX/ESXi 的 DataStore

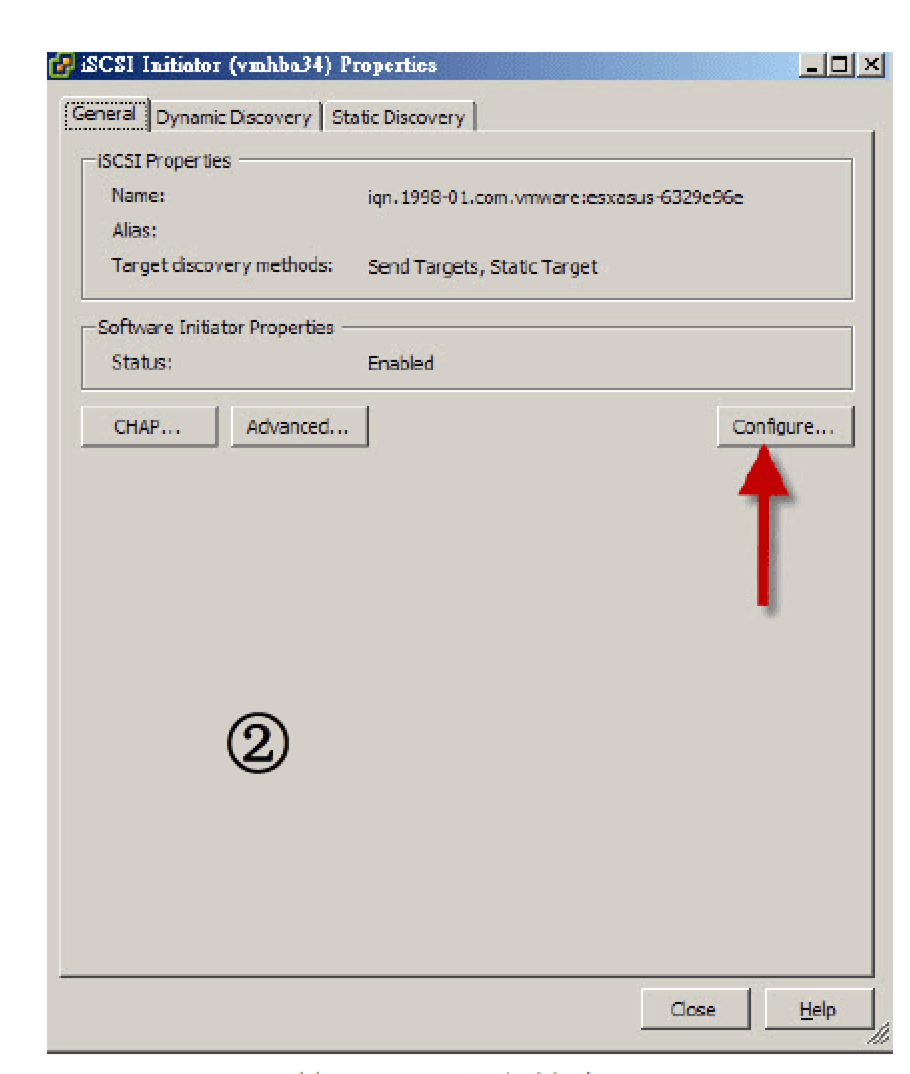
在 ESX/ESXi 中，存储最基本的单位就是 Datastore。由于本例是使用 iSCSI，因此我们的任务主要就是给定 iSCSI Target Server 的 IP 位置即可。

▶ 将 ESX/ESXi 连入 QNAP

1. 进入 vSphere Client 中的 Configuration，选择 Storage Adapters 我们会看到当前拥有的 iSCSI Software Adapter。单击 Properties 按钮。
2. 接着单击 Configure...按钮。

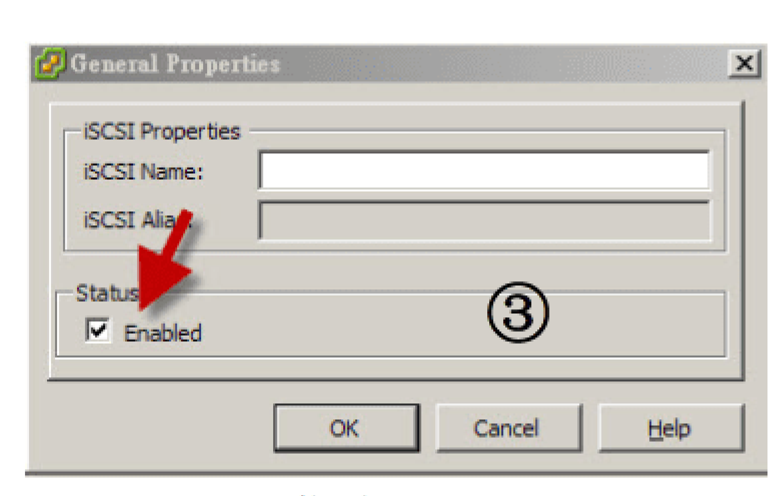


▲ 选择属性的地方

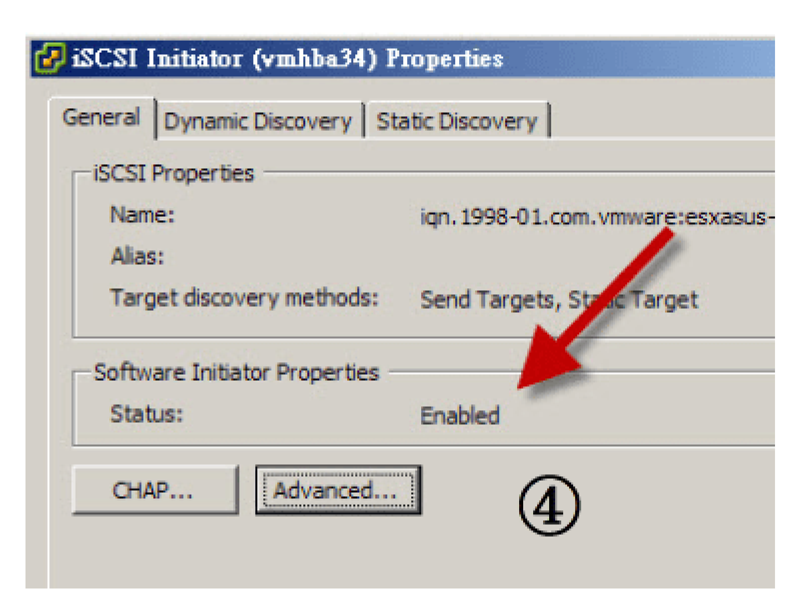


▲ 按下配置的按钮

3. 在下图所示的窗口中选择 Enabled 复选框，单击 OK 按钮离开。
4. 此时我们可以看到 iSCSI 已经激活了。



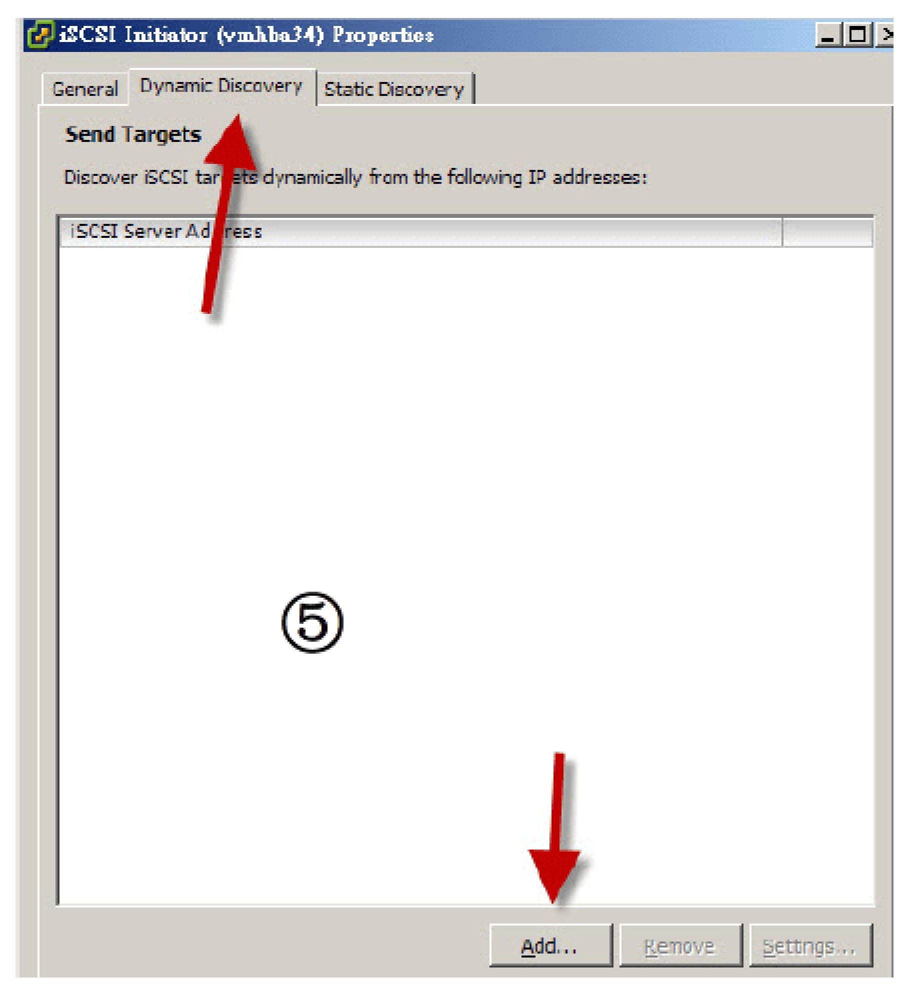
▲ 激活 iSCSI



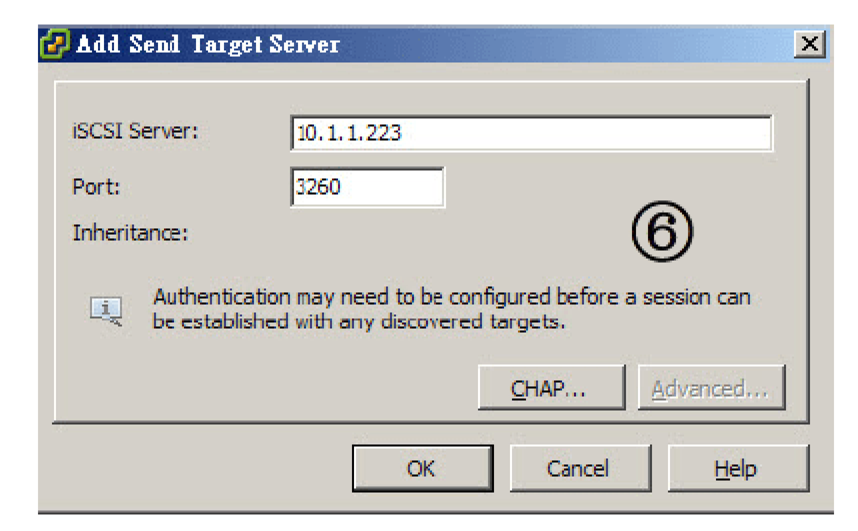
▲ 已激活

5. 选择 Dynamic Discovery 选项卡并且单击 Add…按钮。

6. 键入安装 QNAP 的 IP，本例为 10.1.1.223，之后单击 OK 按钮。



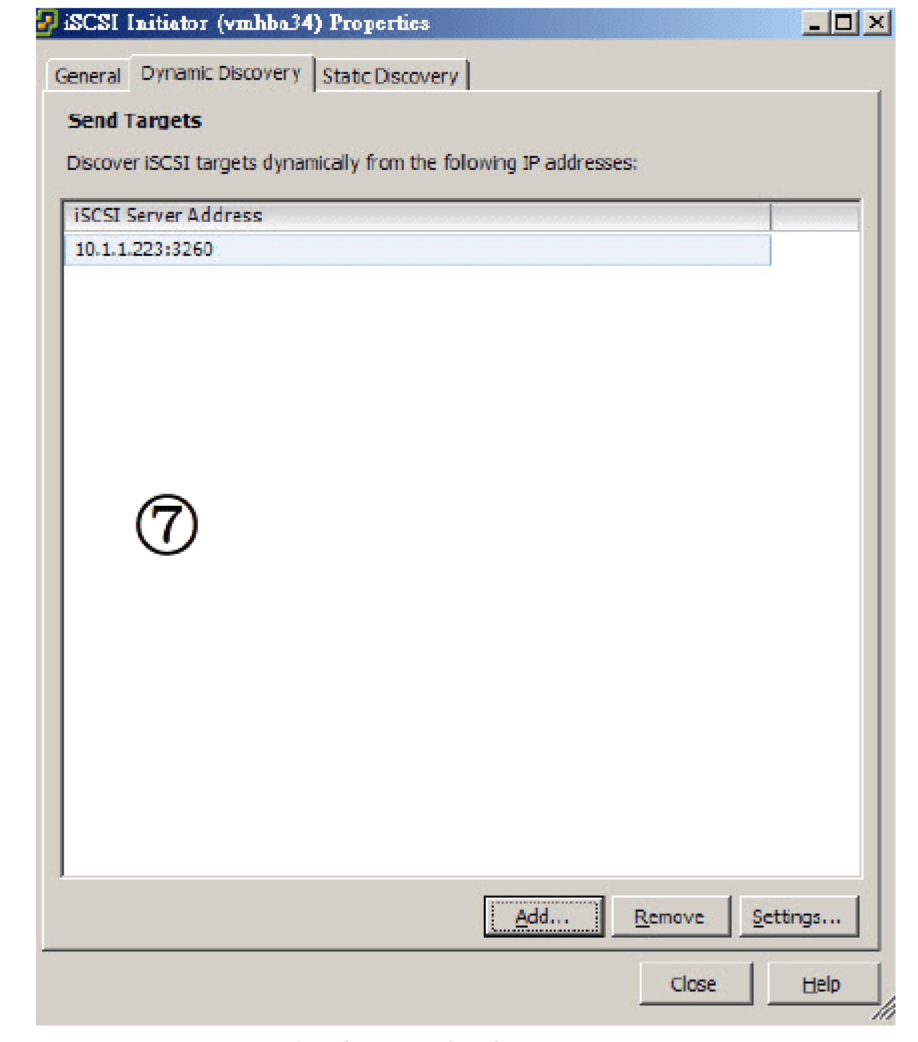
▲ 选择新建 LUN



▲ 键入 QNAP 的 IP

7. 此时我们已经可以看到这个 iSCSI Server 已经被加入了。

8. 由于新加入了 iSCSI，因此系统会要求重新扫描，单击“是（Y）”按钮。

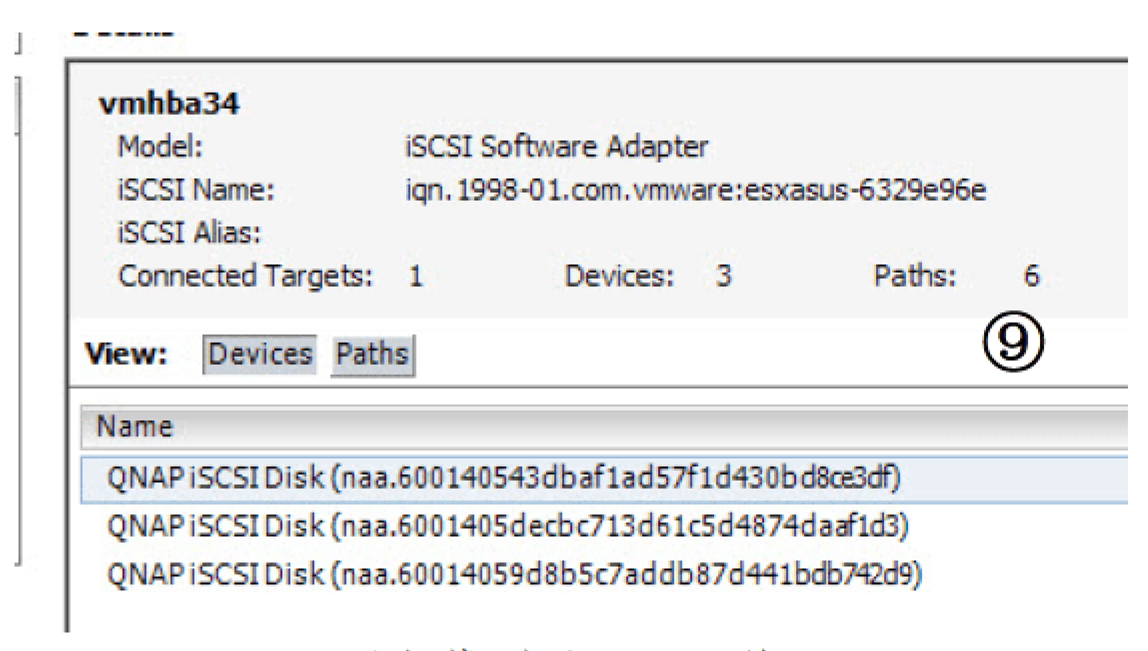


▲ 已经加入这台 Server



▲ ESX/ESXi 会在有新的设备下要求重新扫瞄

9. 你会发现在 QNAP 中的 LUN 已经被对应到 ESX/ESXi 上了，意味着这个 LUN 已经可以被 ESX/ESXi 作为 Datastore。



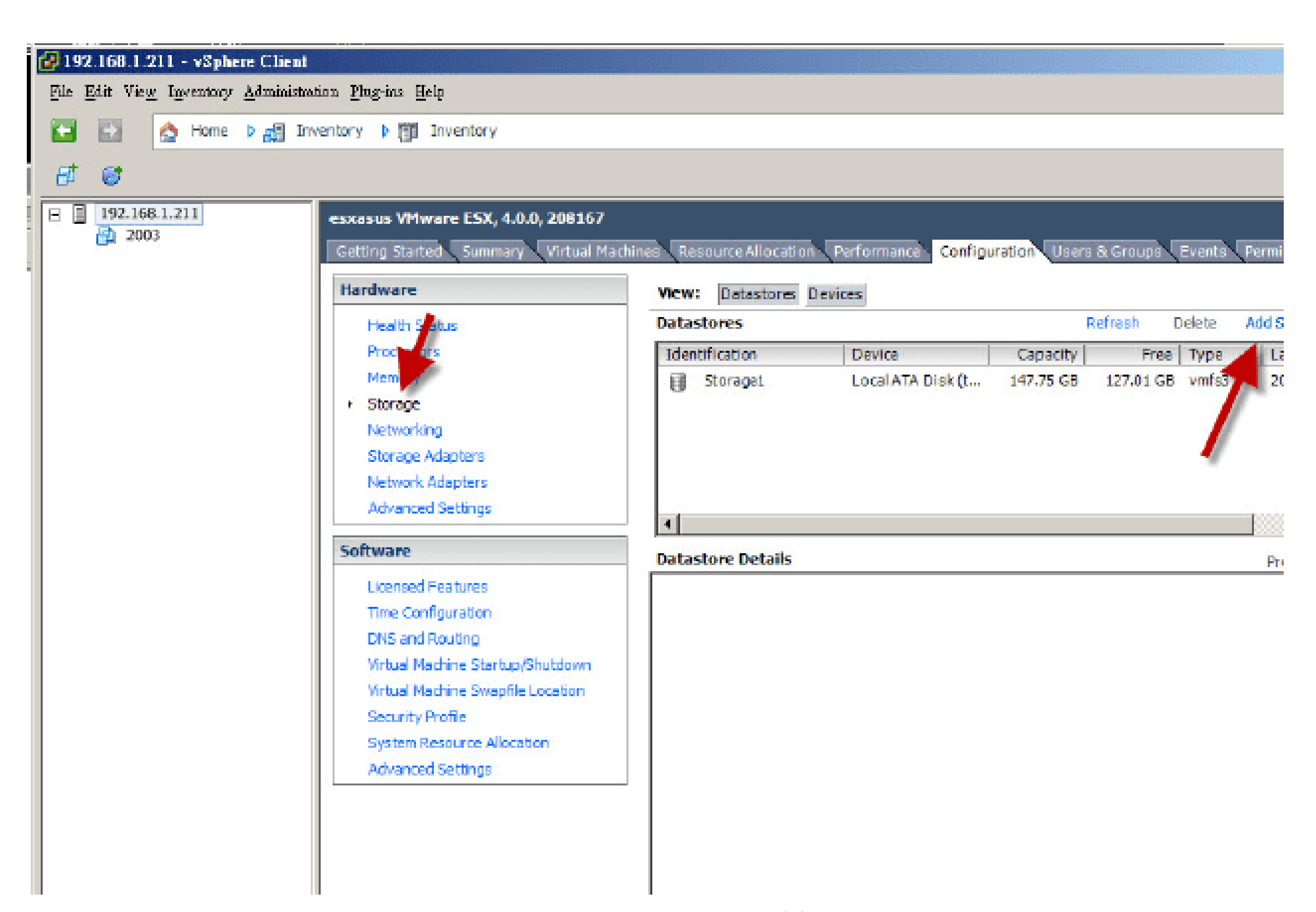
▲ 已经找到了 QNAP 的 LUN

2. 将 iSCSI 的 LUN 加入 Datastore

当 ESX/ESXi 和外部存储设备创建连接之后，我们就可以将这个存储加入 ESX/ESXi 的 Datastore ，而在成为了 Datastore 之后，必须先格式化成 VMFS，之后才能创建虚拟机和上传 ISO 文件。

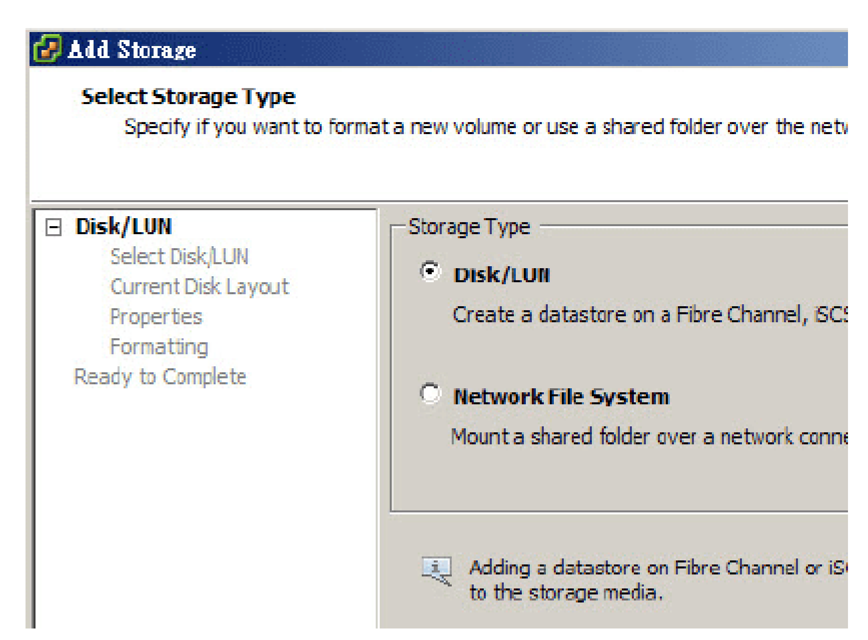
▶ 将 iSCSI 的 LUN 加入 Datastore

1. 进入 vSphere Client 中的 Configuration / Storage 选项，单击 Add Storage。



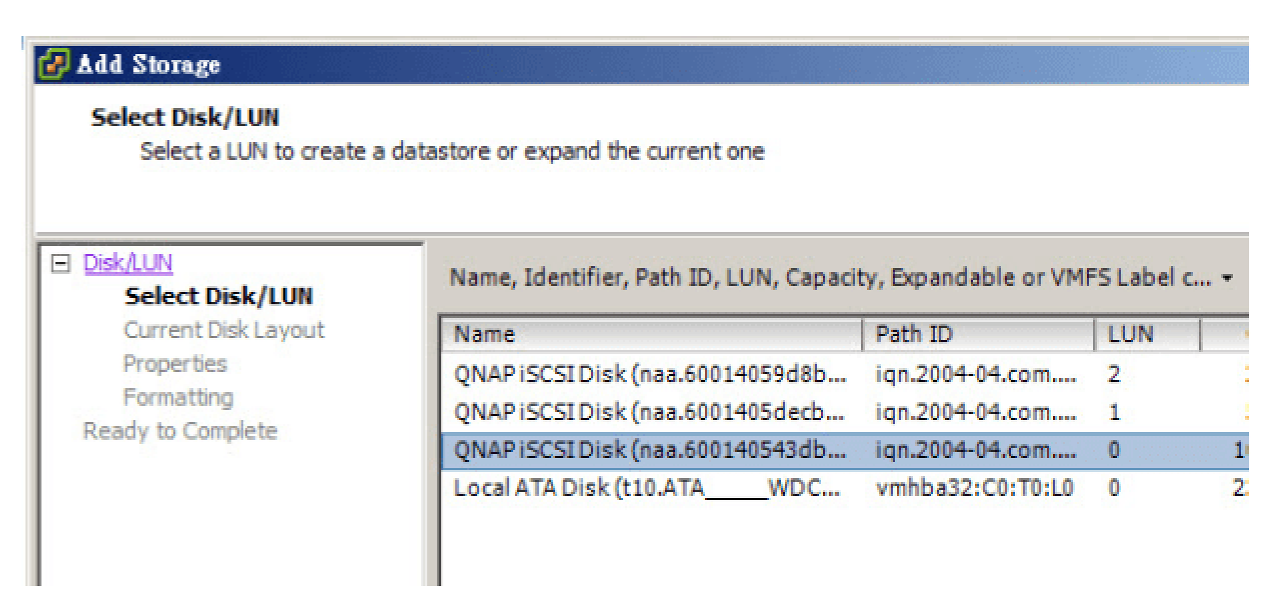
▲ 新添加存储

2. 此时我们选择 Disk/LUN 单选按钮，单击 Next 按钮。



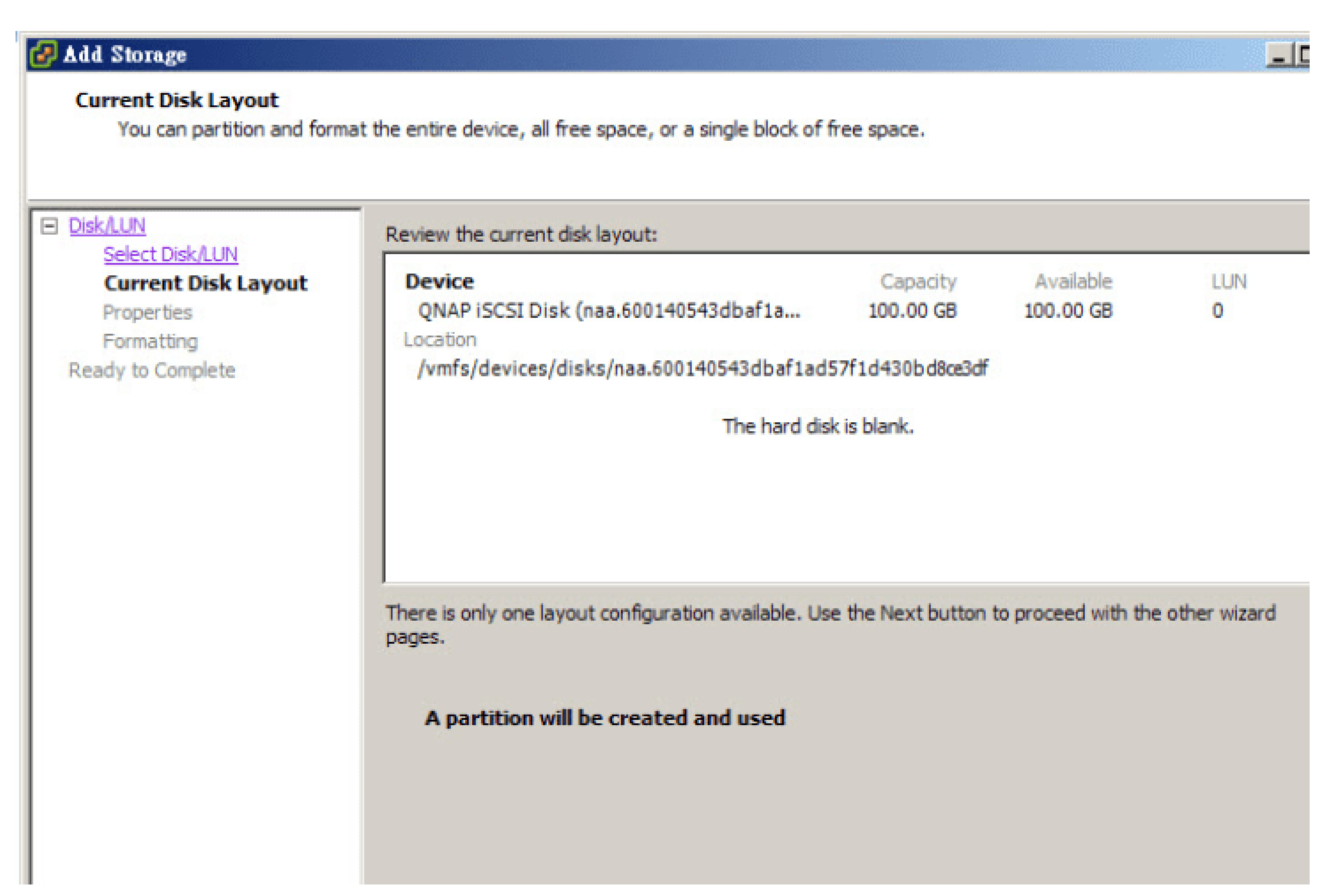
▲ 选择 LUN 这一选项

3. 此时会弹出刚才找到的 iSCSI 硬盘。选择后单击 Next 按钮。



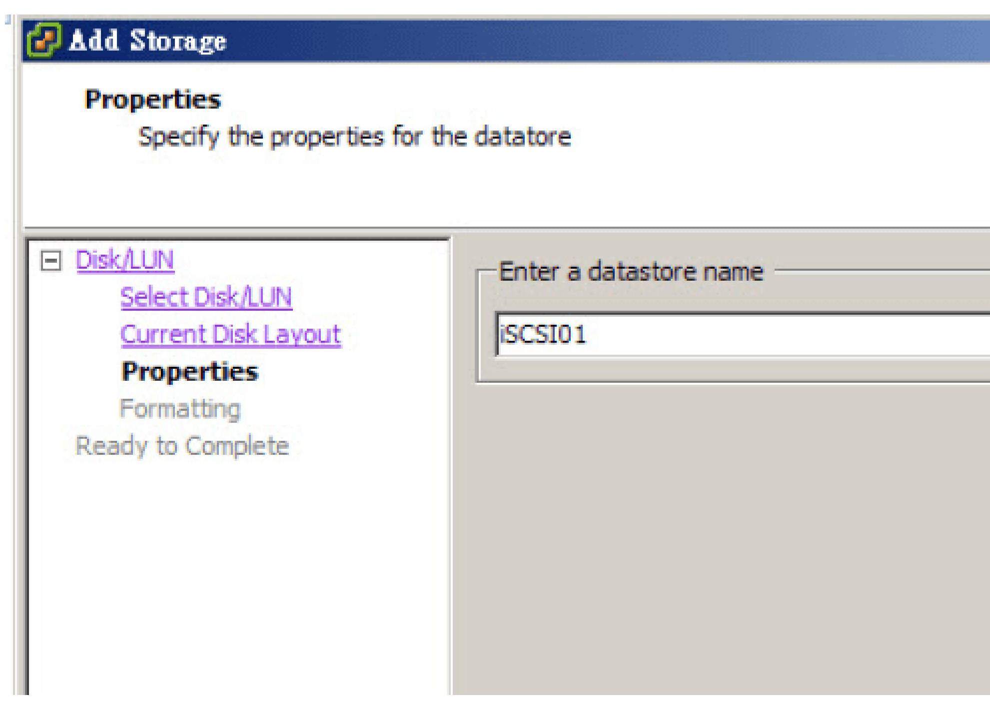
▲ 会弹出刚才找到的 iSCSI

4. 如果这个 LUN 之前被其他的系统使用过，这里会弹出其格式，在加入 ESX/ESXi 之后，上面的资料或文件系统都会被清除。单击 Next 按钮继续。



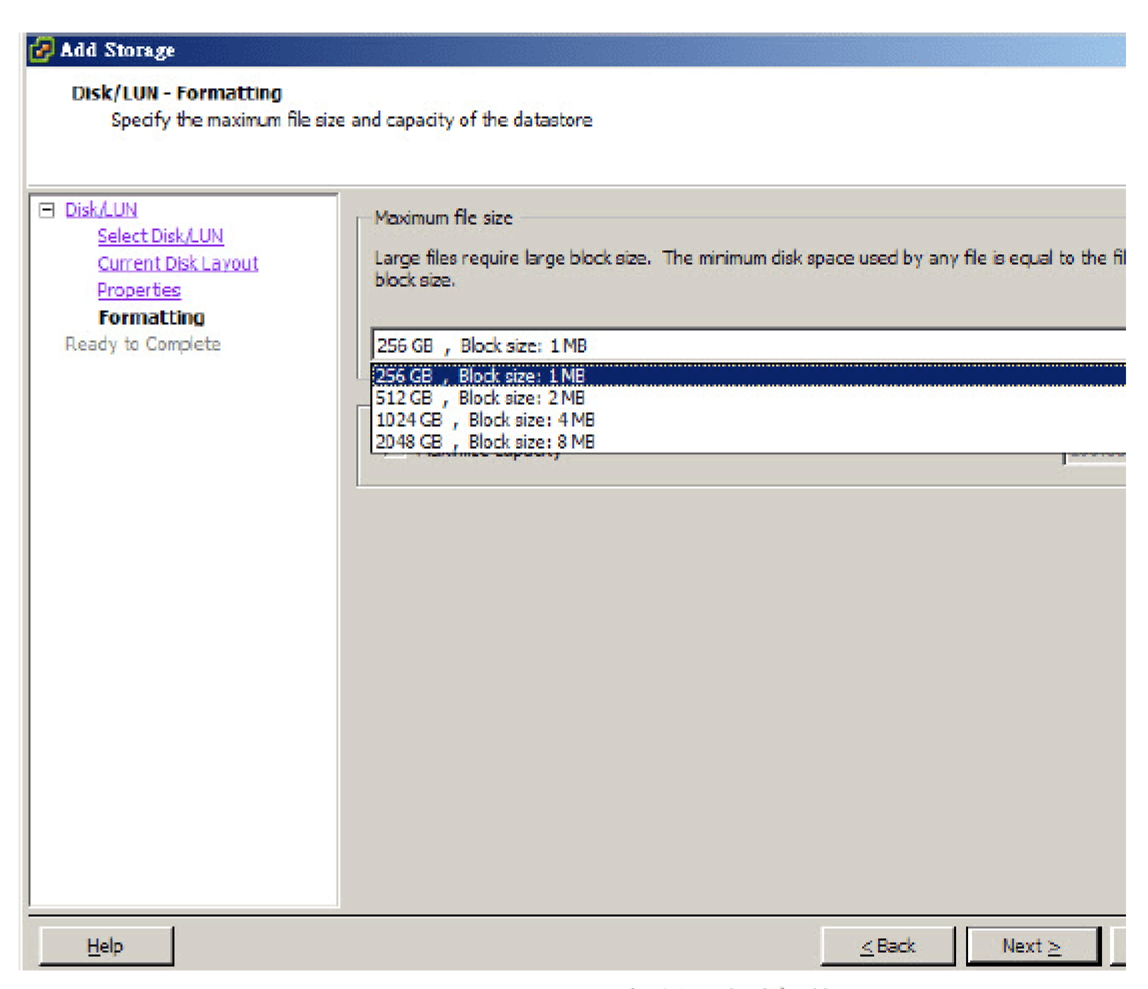
▲ 会被格式化为 VMFS 格式

5. 此时我们替这个 Datastore 取一个名字，如 iSCSI01。



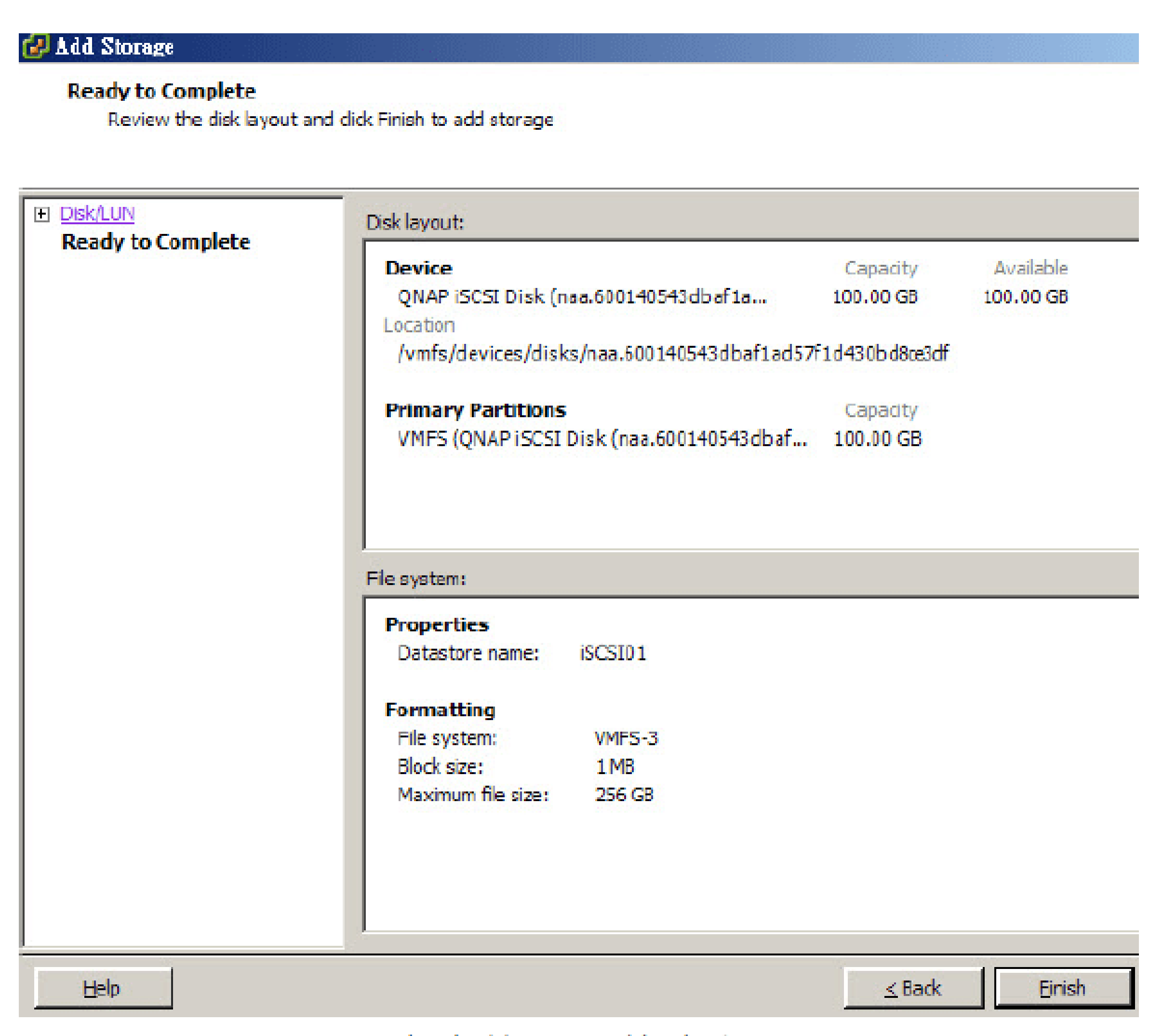
▲ 替这个 Datastore 命名

6. 接下来是配置这个 LUN 的最小访问单位，在 256GB 以下的就以 1MB 作为 Block Size。这里不需要更改，直接单击 Next 按钮继续。



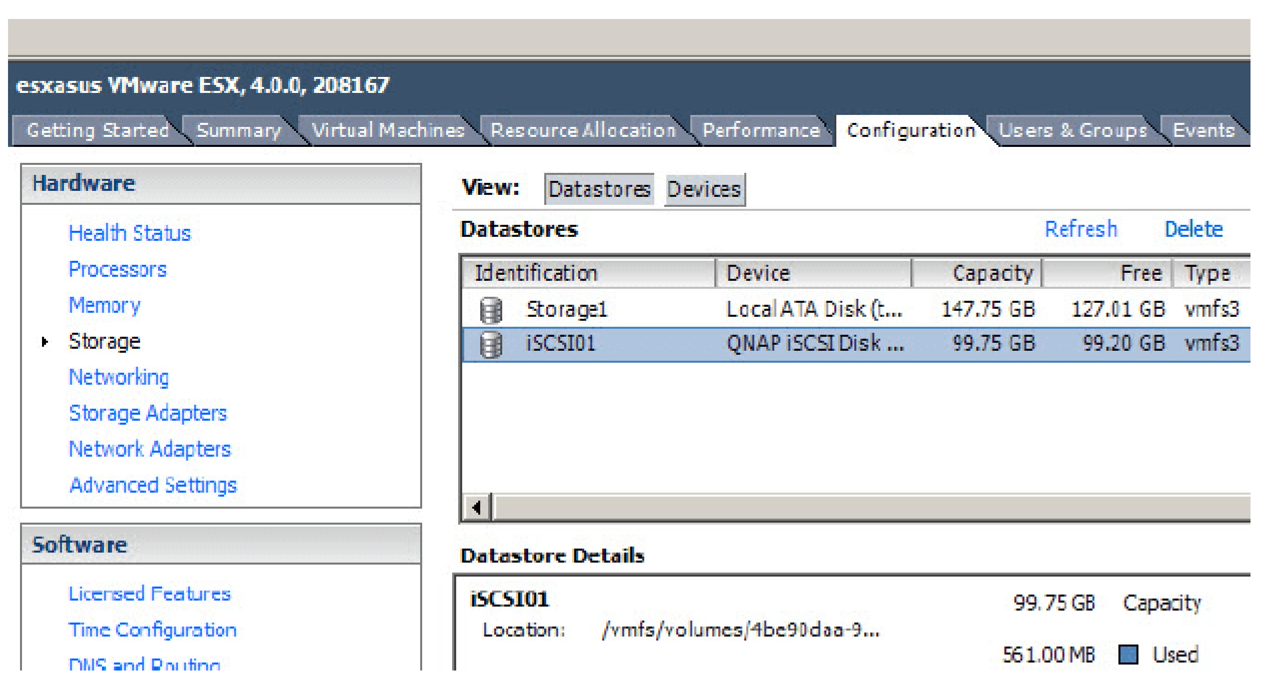
▲ Block Size 会影响效能

7. 此时会弹出这个 Datastore 的完整数据，单击 Finish 按钮就会开始创建。



▲ 会在按下后格式化 LUN

8. 创建完成之后，会回到 Storage 画面，此时会看到创建的 Datastore 已经落实了，而这个 Datastore 也会自动被 ESX/ESXi 格式化为 VMFS 格式。



▲ 落实了



小常识

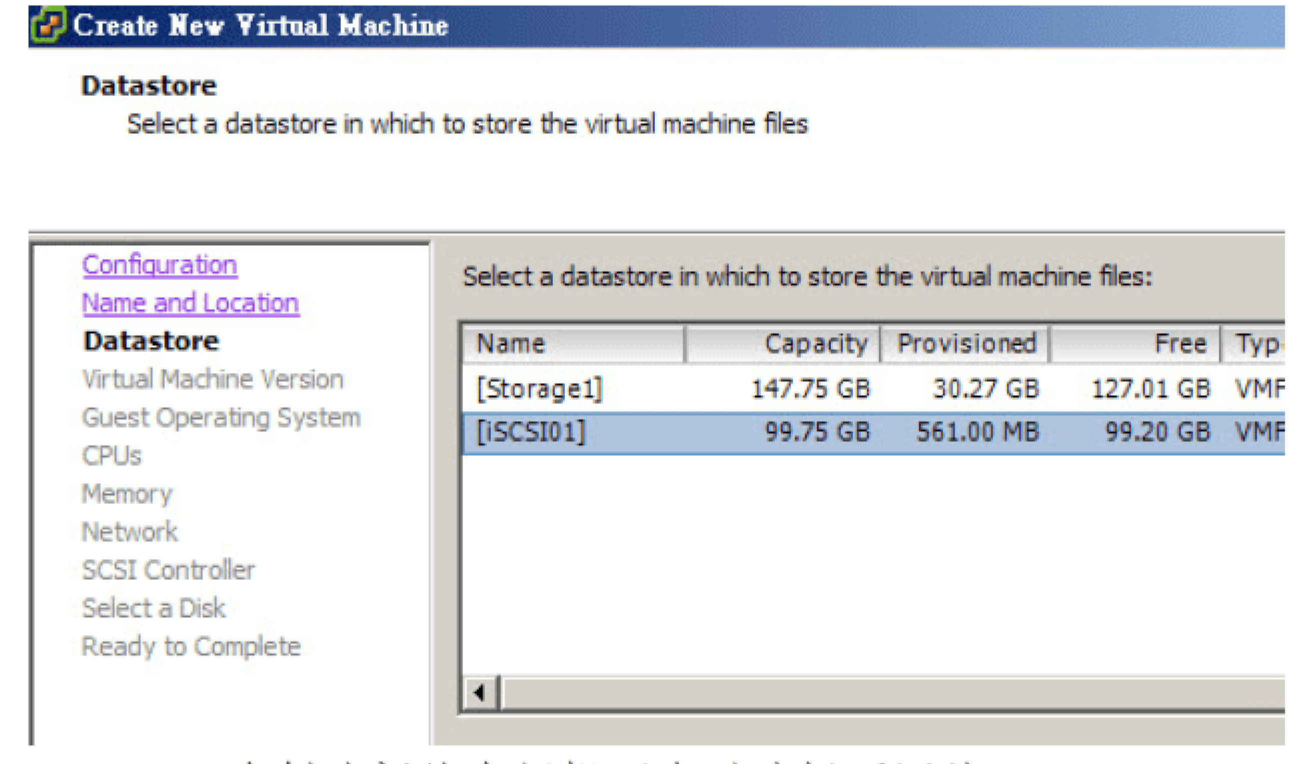
外部存储设备的位置

在 ESX/ESXi 中，我们知道本地硬盘的 Datastore 是存储在/vmfs/volumes 之下，因此在创建了 iSCSI 之后，会在 volumes 之下看到新的 Datastore。而真实的磁盘在 ESX/ESXi 中则是创建于/vmfs/disks 之下。

▲ 在 Linux 中是放在/vmfs 之下

3. 在新的 Datastore 上创建虚拟机

当安装好了新的 Datastore 之后，我们就可以将虚拟机安装在这个 Datastore 上了。安装的方式十分简单，只要在创建虚拟机选择安装位置时，给定安装到新的 Datastore 即可。其他的配置方式则和安装虚拟机到本地硬盘完全一样，我们在此就不详细说明了。



▲ 在创建新的虚拟机硬盘时选择不同的 Datastore

11.3.4 突破 vSphere 的 2TB 制约

vSphere 虽然号称支持高达 64TB 的 Datastore 容量，但这只是理论值。用户常常在这方面遇到问题，下面就来看看如何解决。

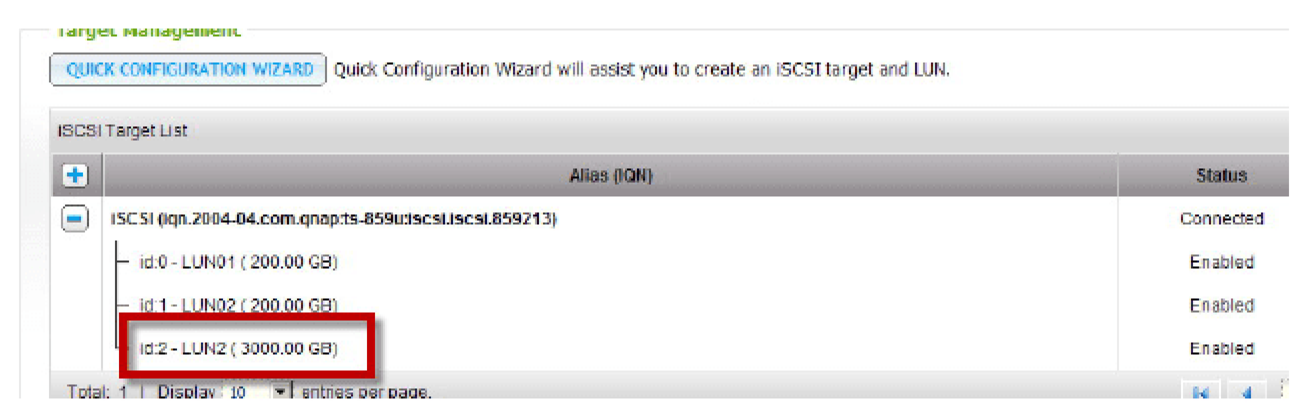
1. 单个 LUN 的制约

事实上 ESX/ESXi 的 Datastore 并无法支持单个硬盘/LUN 的 64TB，因此许多读者在创建 ESX/ESXi 时，发现当本地硬盘大于 2TB 时，Datastore 竟然只能使用总容量-2TB。举例来说，如果你创建一个 2.7TB 的 Datastore，会发现可用容量只有 700GB。因此如果有超过 2TB 的磁盘或是 LUN，就无法使用传统的方式了。

2. 加入 2TB 以上容量的 Datastore 做法

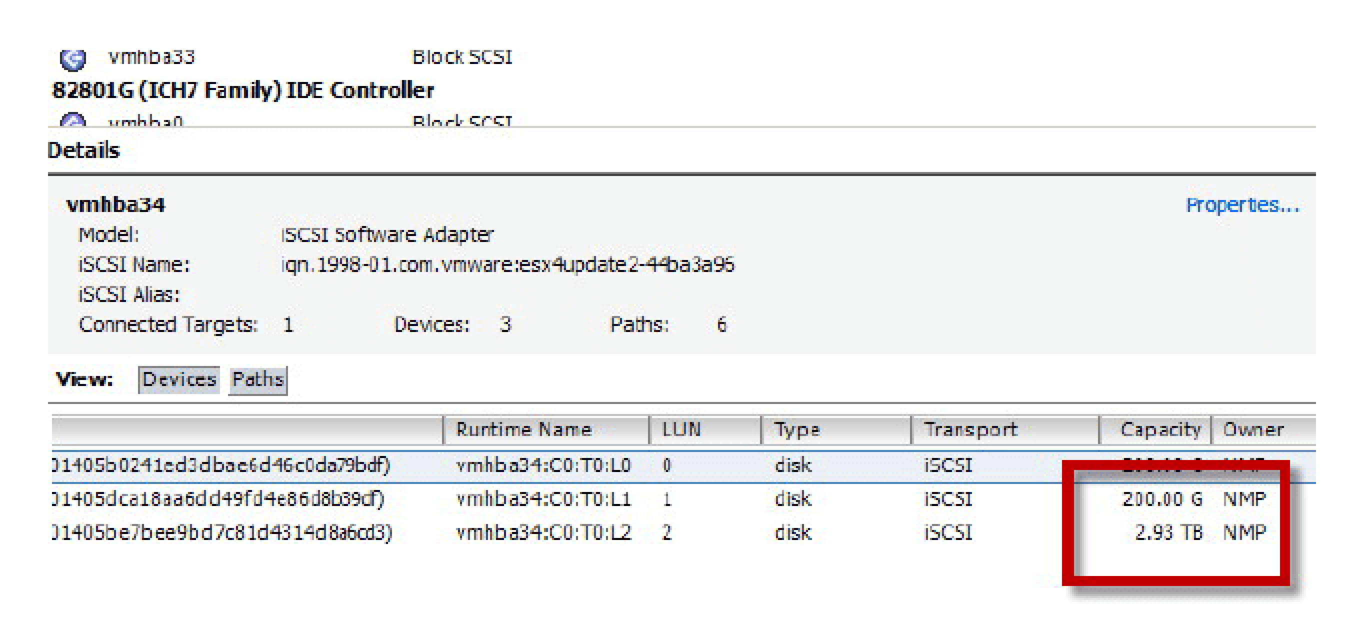
大部分的用户会用以下的做法加入 Datastore，但超过 2TB 时就会有问题，我们就来看看。

1. 首先进入 NAS 的配置界面，创建一个超过 2TB 的 LUN，如 2.93TB 的一个 LUN2。



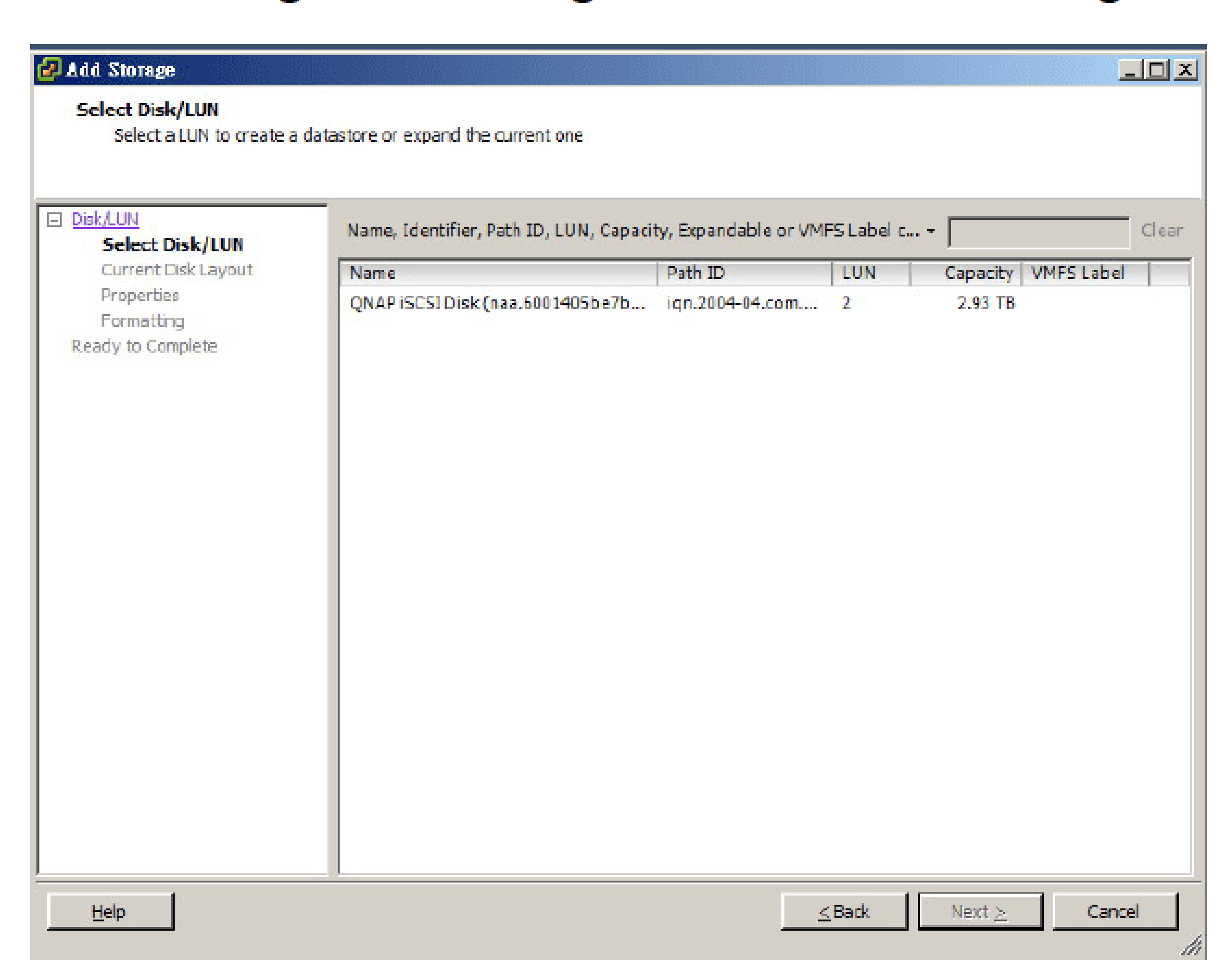
▲ 先在 NAS 中创建一个 3TB 的 LUN

2. 创建完之后，回到 ESX/ESXi 主机下，并且进入 Configuration/Storage Adapter，选择 Rescan 重新扫描。此时我们可以发现在这个 Storage Adapter 下，可以发现 2.93TB 的 LUN。



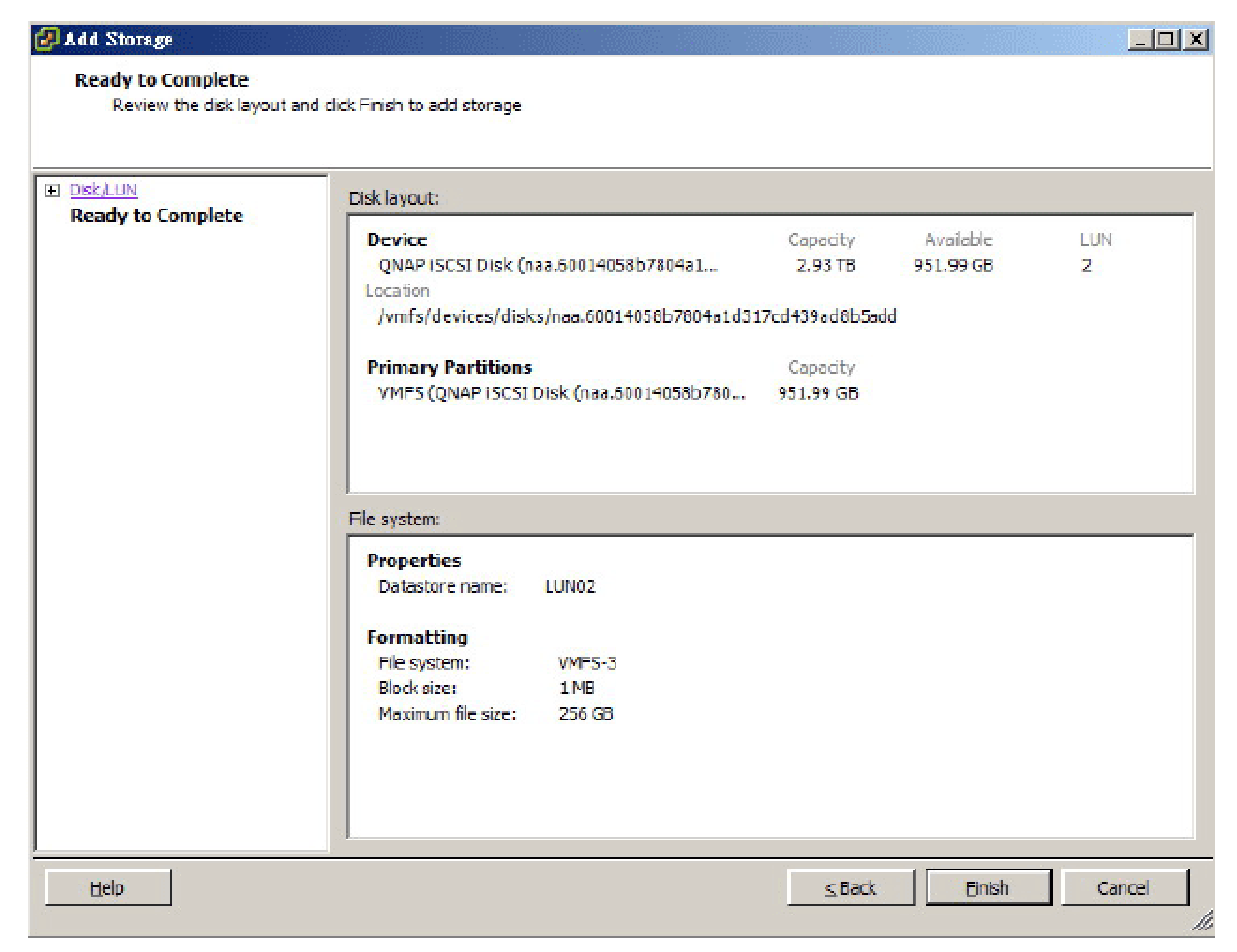
▲ 看到这个 LUN 了

3. 此时我们回到 Configuration/Storage，并且单击 Add Storage。



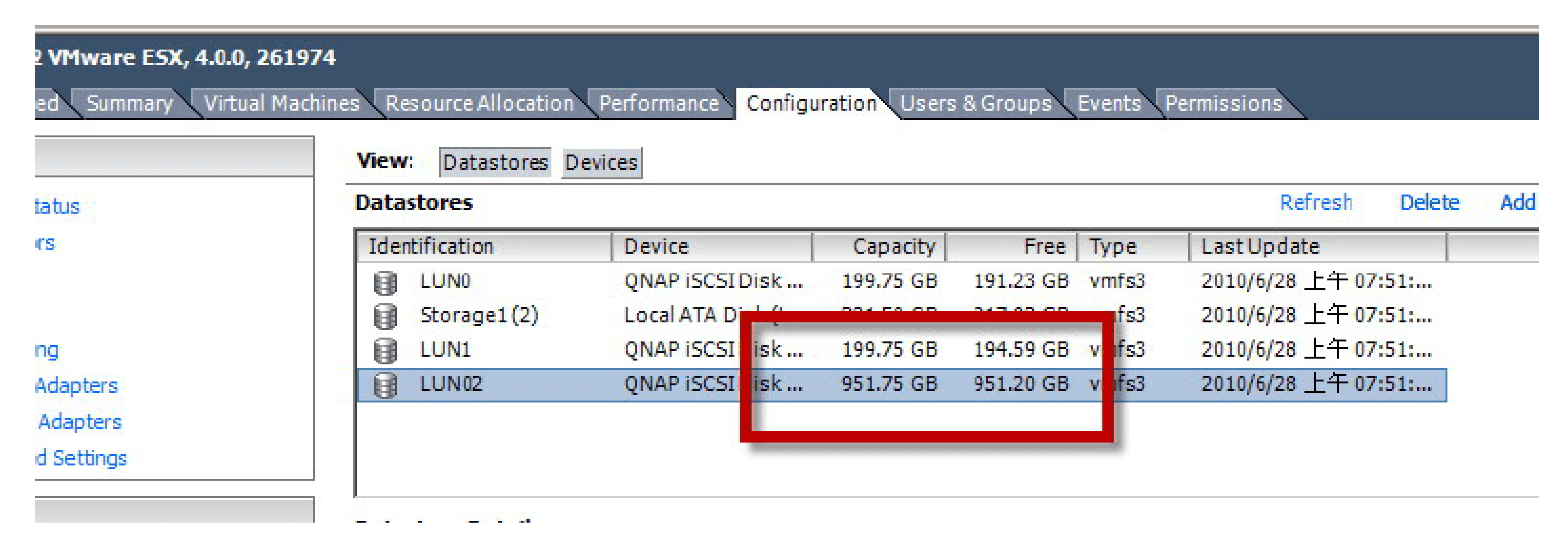
▲ 加入新的 Storage

4. 此时我们选择 LUN，可以看到那个 2.93TB 的 LUN。之后将其加入，但会发现其可用容量是 983GB，即 2.93TB 减去 2TB 的容量。



▲ 虽然可以加入，可用容量只有 0.93TB

5. 创建好的 Datastore 只有 0.93TB，是无法使用的。



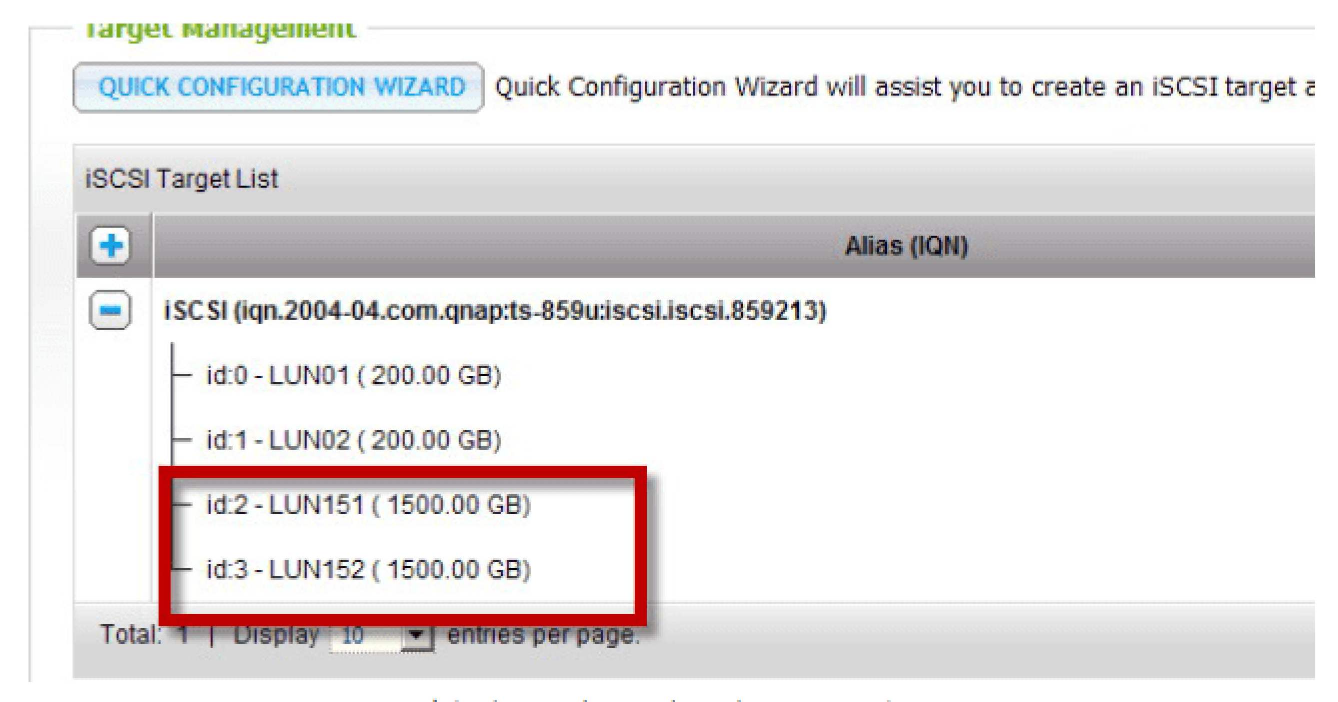
▲ 创建好的也只能用到 0.93TB

3. 使用扩展容量的方式

在 ESX/ESXi 中，单个 LUN/硬盘无法超过 2TB，但可以用扩展容量的方式让一个 Datastore 有多个 LUN/硬盘。这种混合式的做法对用户来说是完全透明的，当前支持的做法是可以使用本地硬盘和 LUN（不管是 FC SAN 或是 iSCSI），还不支持 NFS 加入。

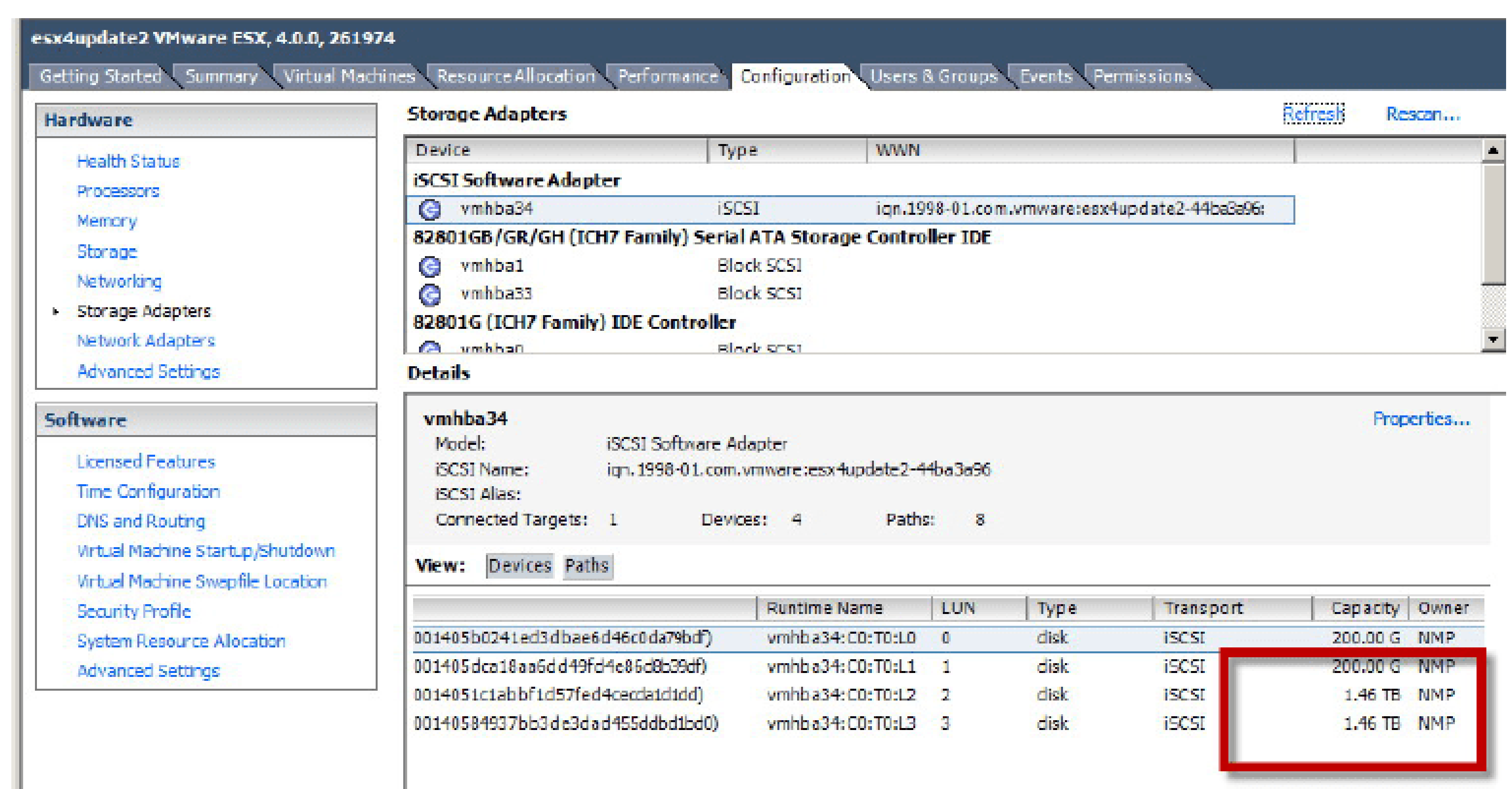
▶ 让 Datastore 容量超过 2TB 的做法

1. 首先我们要确定创建的 LUN 单个容量不可超过 2TB，因此以 3TB 来说，我们可以创建两个 1.5TB 的 LUN。



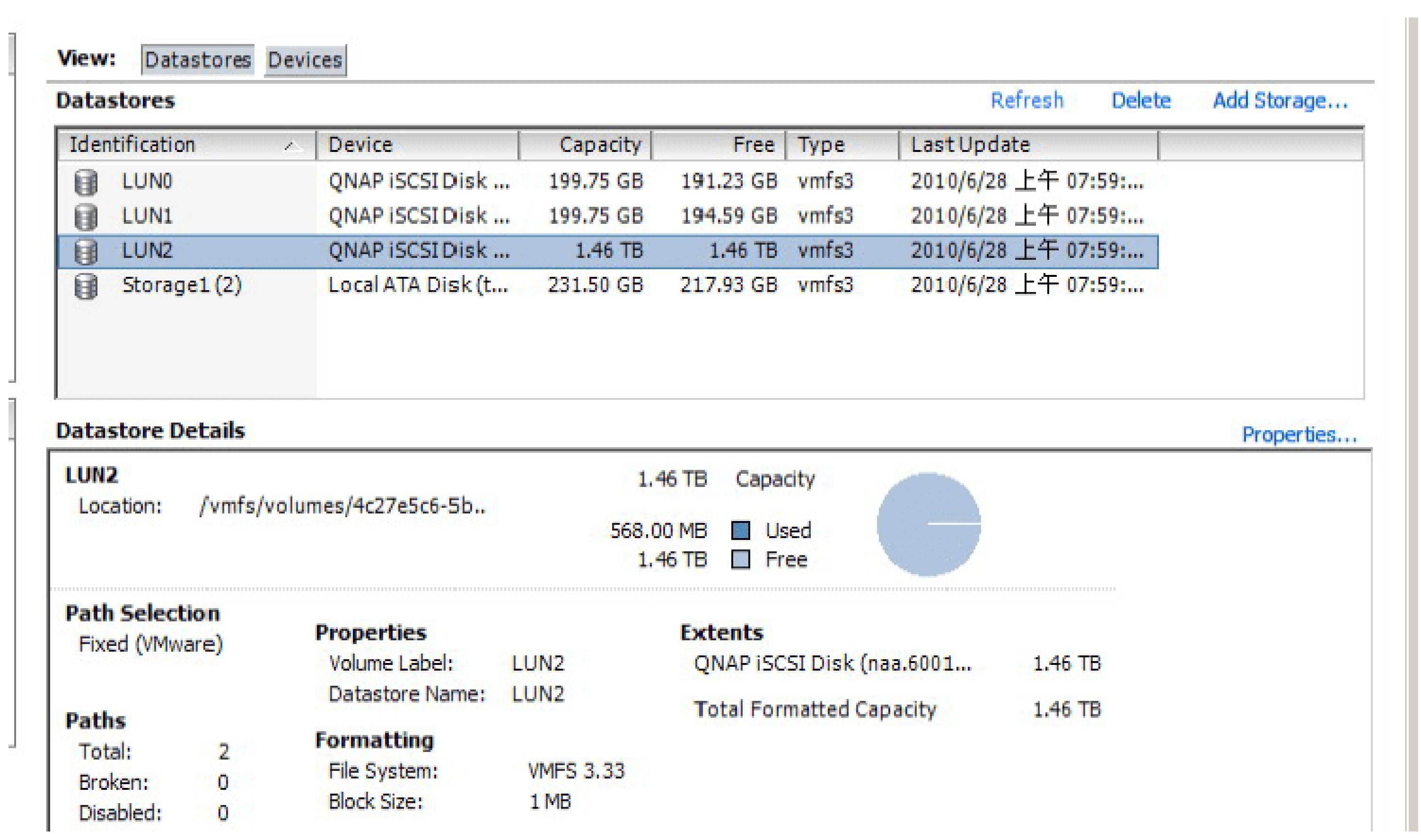
▲ 创建两个不超过 2TB 的 LUN

2. 进入 ESX/ESXi 中，而且确定已经找到这两个 LUN。



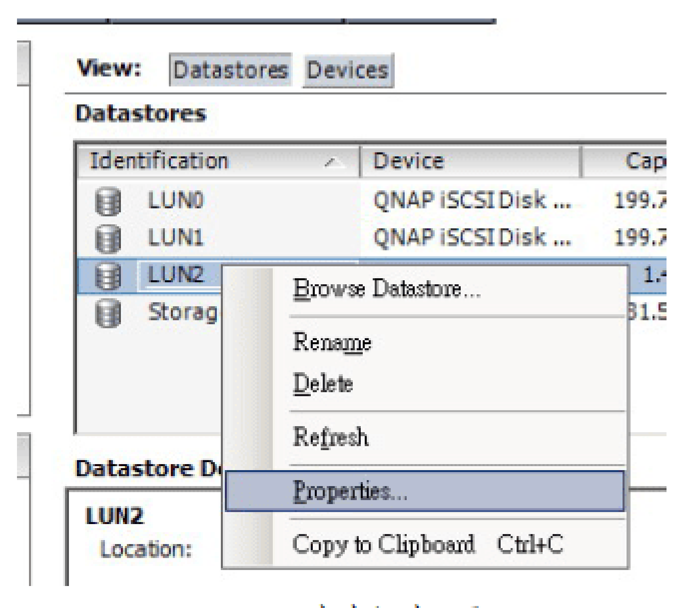
▲ 确定找到这两个 LUN

3. 先用正常的方式将第一个 1.5TB 的 LUN 加成 Datastore。如下图所示的窗口中已经加成 LUN2 了。



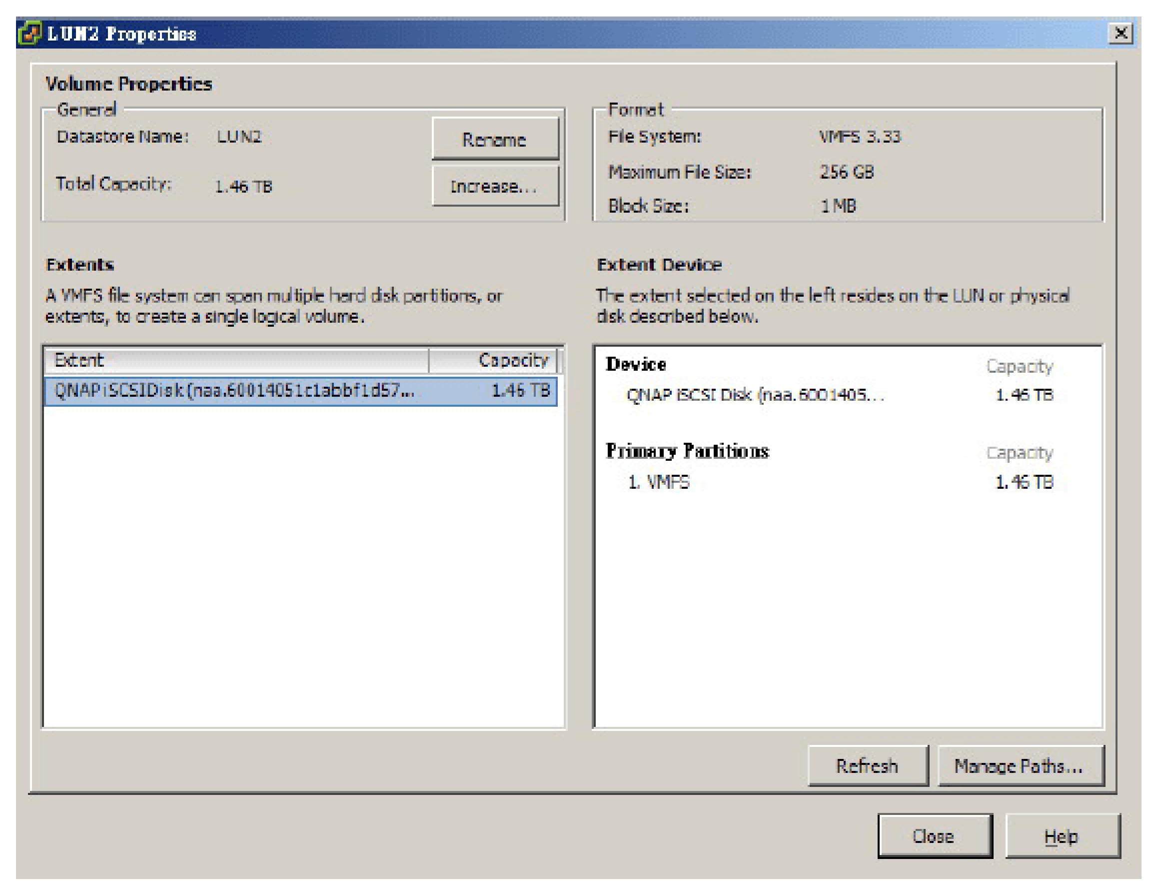
▲ 先将第一个 LUN 加入 Datastore

4. 在该 Datastore 上右击，选择 Properties…选项。



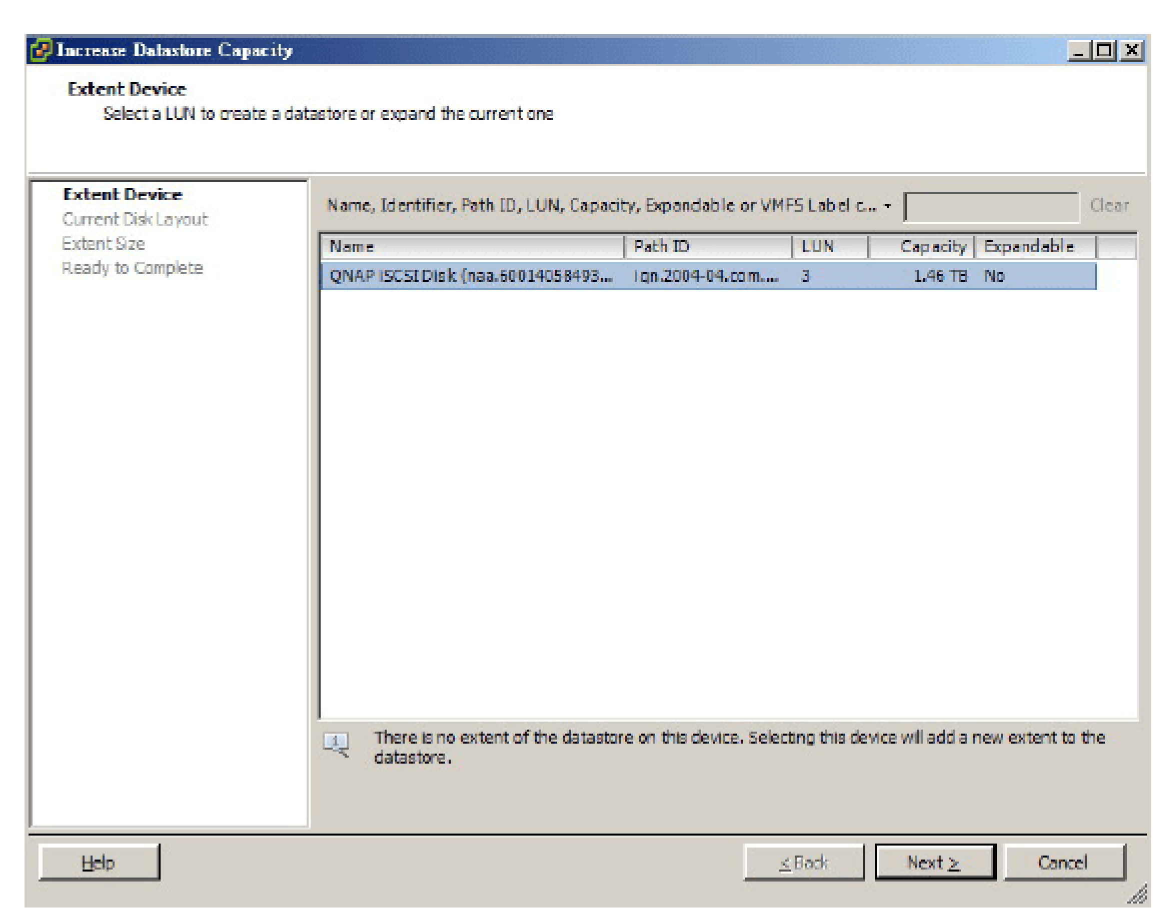
▲ 选择选项

5. 此时我们可以在下图所示的窗口中看到这个 Datastore 的基本数据。在图中的 Extents 列表框，就是这个硬盘可以扩展的其他 LUN/本地硬盘。我们直接单击 Increase 按钮来添加。



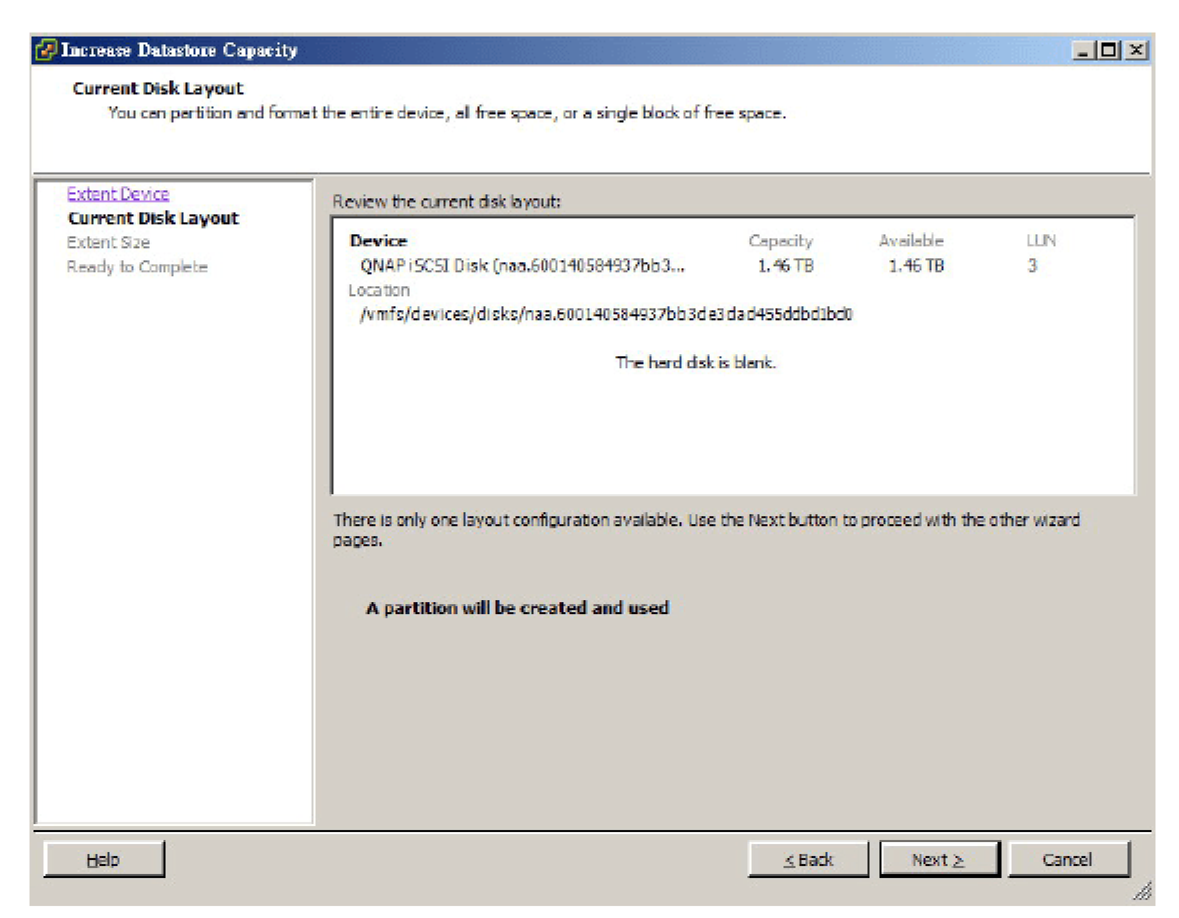
▲ 添加容量

6. 此时会给出可以扩展的 LUN 或硬盘。选择之后单击 Next 按钮继续。



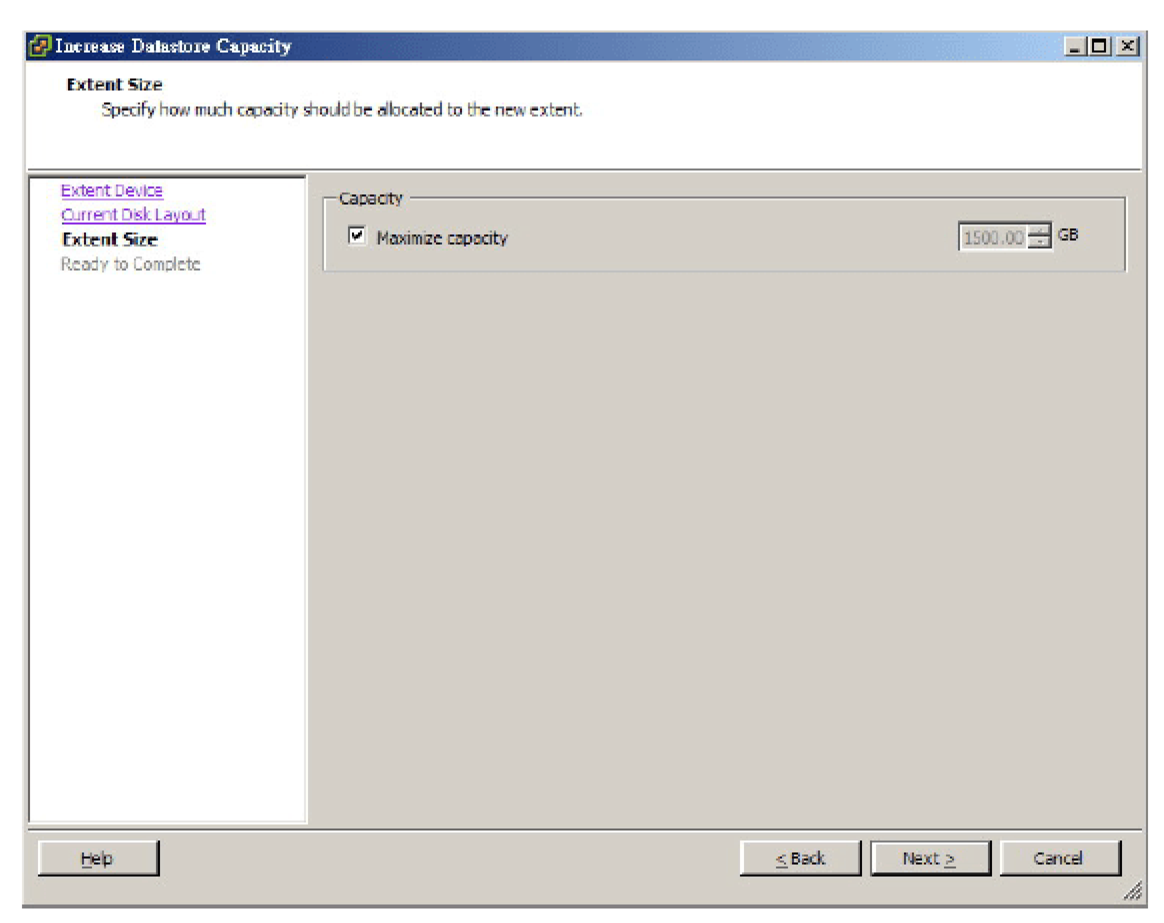
▲ 选择第二个 1.5TB 的 LUN

7. 接下来会给出该 LUN 的总结，单击 Next 按钮继续。



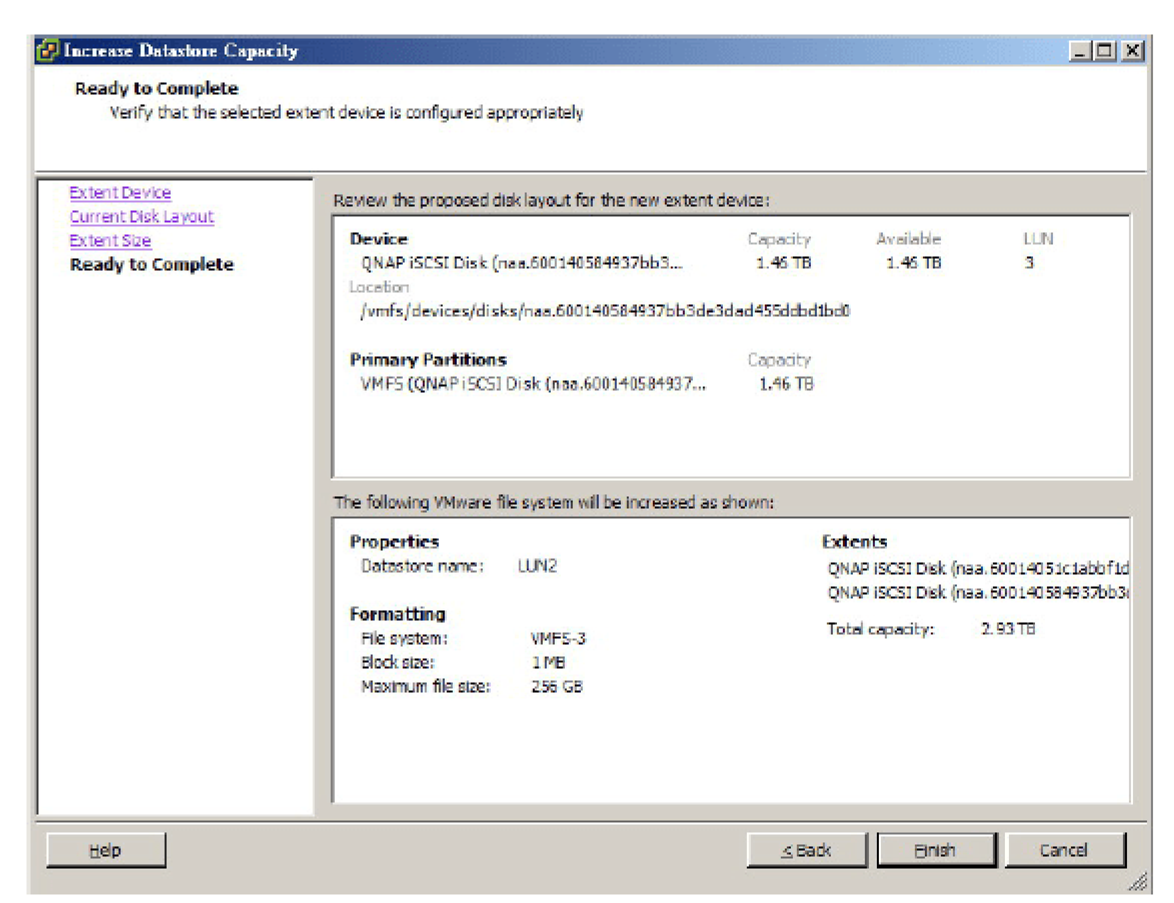
▲ 找到这个了，可以加入

8. 此时会给出需要的容量，我们就选择最大容量，单击 Next 按钮继续。



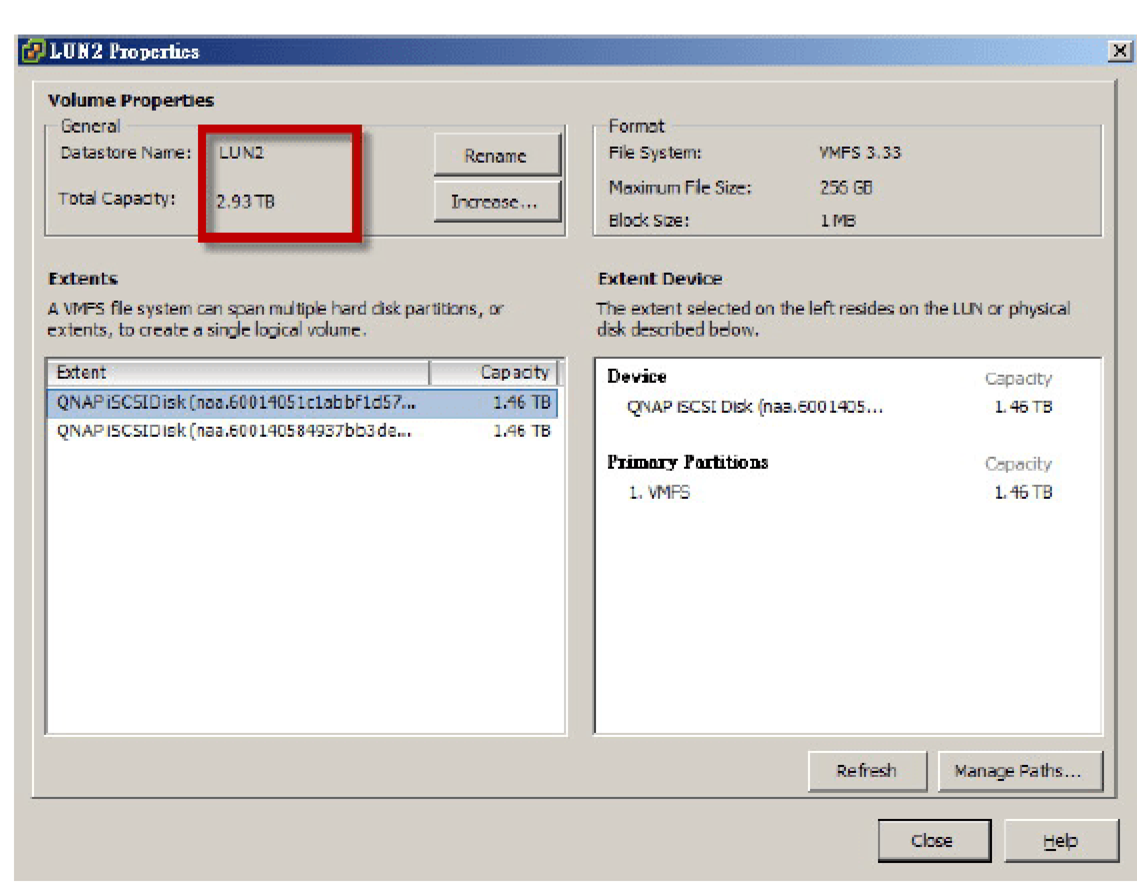
▲ 选择最大容量

9. 接下来会有加入的总结，单击 Finish 按钮继续。



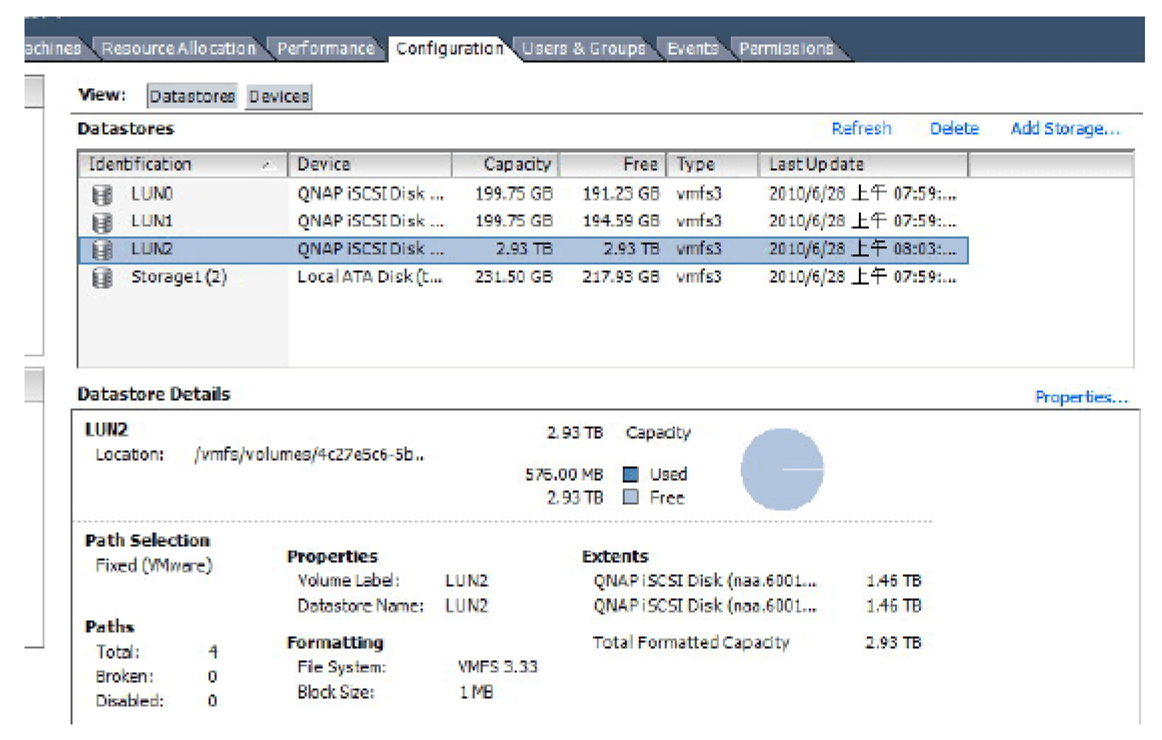
▲ 加入总结

10. 当落实后，我们可以看到这个 Datastore 有两个 Extents，容量为 2.93TB（即 3TB）。



▲ 两个 LUN 组合成一个 Datastore

11. 在 Datastores 视图中，也看到是超过 2TB 的容量了。



▲ 容量正常了

结 语

本章说明了 ESX/ESXi 中最重要的存储观念，虽然介绍简单，但足以应付大部分单机的 ESX/ESXi 虚拟机存储应用了。存储是虚拟化的基础，理解 ESX/ESXi 下的存储设备之后，接下来就是准备好好应用这些外部的存储设备了。我们在下一章会说明更多有关存储的应用，读者们要将这一章的基础打好。

第 12 章 使用 OpenFiler 自制虚拟机环境的 SAN

关键词：

- 自制 iSCSI 的 SAN 存储设备——使用 OpenFiler
  - OpenFiler 是一个免费的 NAS/iSCSI 的 SAN 服务器操作系统，提供 LAN 主机的独立存储系统，使用 iSCSI Target
  - OpenFiler 目前最新的版本为 2.3，在官方网站上提供了原始安装的 ISO 档和已经安装完毕的虚拟机使用
  - 将 OpenFiler 安装到实体机上就和安装任何一个 Linux 操作系统一样
- 使用 OpenFiler 基本功能
  - 了解基本操作，如开关机、配置密码及 IP、主机名等
  - 创建实体硬盘
  - 创建 PV 的步骤
  - 创建 VG 的步骤
  - 创建 LUN 的步骤
  - 创建 iSCSI 的对应
  - 使用 XP 测试

在上一章我们介绍了存储设备中最重要 VG 原理和 iSCSI 的架构，但在虚拟机环境正式使用之前，独立的存储设备是最基本的，然而大部分的公司也不太可能在评估设备之前就直接购买昂贵的 SAN 设备。OpenFiler 的出现解决了这个问题，让企业在真正采购 SAN 或是高价的 HBA 之前，能使用 iSCSI 来完成虚拟机存储设备这一部分，甚至可以将 OpenFiler 用在生产环境，我们在这一章就来看看 OpenFiler 完整的安装、配置及使用。

12.1 自制 iSCSI 的 SAN 存储设备——使用 OpenFiler

大部分的存储设备上都有自己的系统来进行 RAID 或 VG 的配置工作，这些存储设备使用的是厂商自己开发的专属系统。随着 PC 服务器的流行，许多厂商也开始开发通用的操作系统将整个 PC 转变成存储设备，本小节要介绍的就是目前在虚拟机业界最好用的 OpenFiler。

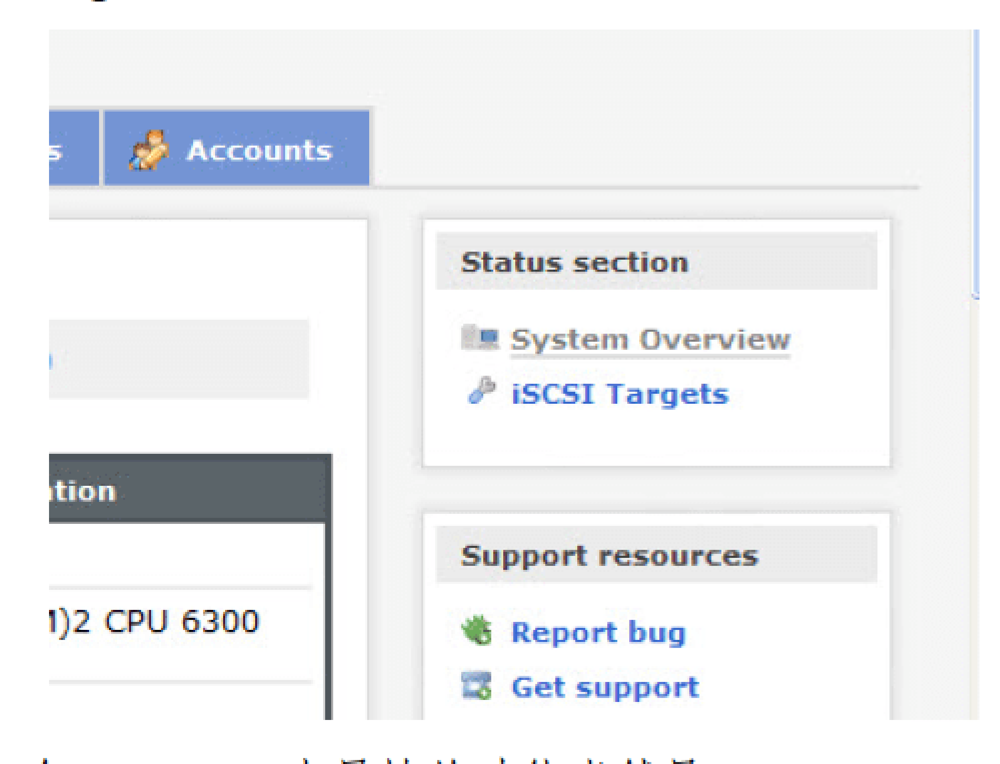
Manage Services

Service Name	Status	Modification
SMB / CIFS server	Enabled	Disable
NFSv3 server	Enabled	Disable
HTTP / WebDAV server	Disabled	Enable
FTP server	Enabled	Disable
iSCSI target server	Enabled	Disable
Rsync server	Disabled	Enable
UPS server	Disabled	Enable
LDAP server	Disabled	Enable
ACPI daemon	Enabled	Disable
iSCSI initiator	Enabled	Disable

▲ 使用操作系统还可以安装不同的分享方式，如 NFS 或 FTP

12.1.1 OpenFiler 的简介

OpenFiler 是一个免费的 NAS/iSCSI 的 SAN 服务器操作系统，主要的目的就是提供 LAN 主机的独立存储系统。OpenFiler 提供了 iSCSI Target，并且有 NFS 文件系统、SMB 文件系统（给 Windows 系统使用），更可以使用 HTTP 来分享文件。主机可以使用 iSCSI Initiator 来做直接连接，也可以使用共享文件夹或是挂载的方式来访问 OpenFiler 上的数据，对于想要进行虚拟化评估的中小企业来说，是 Provisioning 阶段最好用的工具。



▲ 在 OpenFiler 中最棒的功能当然是 iSCSI Target Server

12.1.2 OpenFiler 的下载及安装

OpenFiler 目前最新的版本为 2.3，在官方网站上提供了原始安装的 ISO 档和已经安装完毕的虚拟机使用，包括 VMware 的 Workstation、Server 和 ESX Server、Xen、Virtual Iron 和 Parallels 等虚拟机产品。如果你想将 OpenFiler 安装在一台有多个硬盘的机器上，可以下载 ISO 来安装，如果你只想测试 OpenFiler，可以下载 VMware 的虚拟硬盘直接在 Workstation 版本的 VMware 上直接使用。读者们可以到http://www.openfiler.com/community/download的网址选择需要的版本下载，ISO 文件的大小约 328MB，而 VMware 的压缩文件大小约为 202MB。

Download

The Openfiler distribution is available as an installable ISO CD image to be deployed on bare-metal or a pre-installed disk image for use in one of the several supported virtual machines monitors.

Installable images are available for both x86 and x86-64 architectures. Supported virtual machines monitors include Xen, VMware, QEMU, Virtual Iron and Parallels.

Please download the correct version of Openfiler for your architecture and virtual machine monitor where applicable.

Installation Guides

- Graphical Installation
- Text-based Installation

Openfiler NAS/SAN Appliance, version 2.3 (Final Release)

arch	type	description	download	sha1sum
x86	ISO Image (x86)	Installation ISO image (x86)		8343a9ae07a0590d88484b3dac117c843626bf9c
x86_64	ISO Image (x86/64)	Installation ISO image (x86/64)		dae55d478ade41389cff3718579745b1a95f6d41
x86	VMware Virtual Appliance (x86)	x86 VMware Virtual Appliance		a1d10dfa3fe88675d3bada60d6fa7880c3e9d333

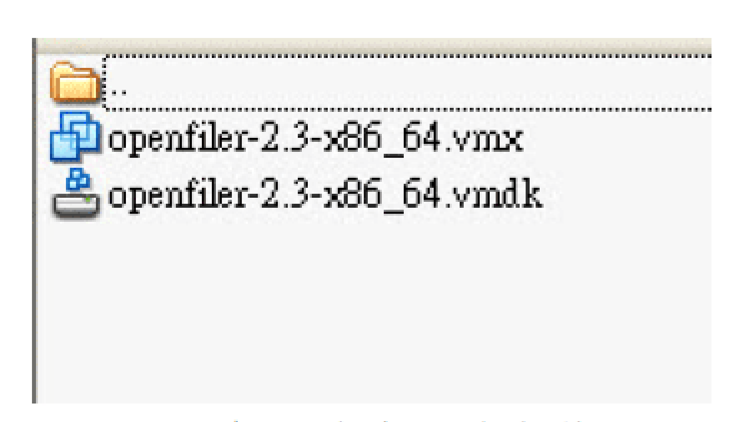
▲ 在此下载 OpenFiler

1. 安装 VMware 版本的 OpenFiler

将 OpenFiler 安装到 VMware ESX Server 或 Hyper-V 下意义并不大，因为我们使用 OpenFiler 的目的就是要给这些虚拟机产品使用。一般来说，如果你的目的只是测试 SAN 和 iSCSI，我们会将 OpenFiler 的虚拟机安装在 VMware 的 Workstation 这一类寄居架构的虚拟机产品上。

▶ OpenFiler 将安装在 VMware Workstation 或 Server 上

1. 先将下载回来的压缩文件解压，其中包括两个文件的文件夹，一个是 OpenFiler 的硬盘文件，一个是 VMware 的配置文件。
2. 解压缩完毕之后，直接开始以 VMware Workstation 或是 Server 来运行 OpenFiler。



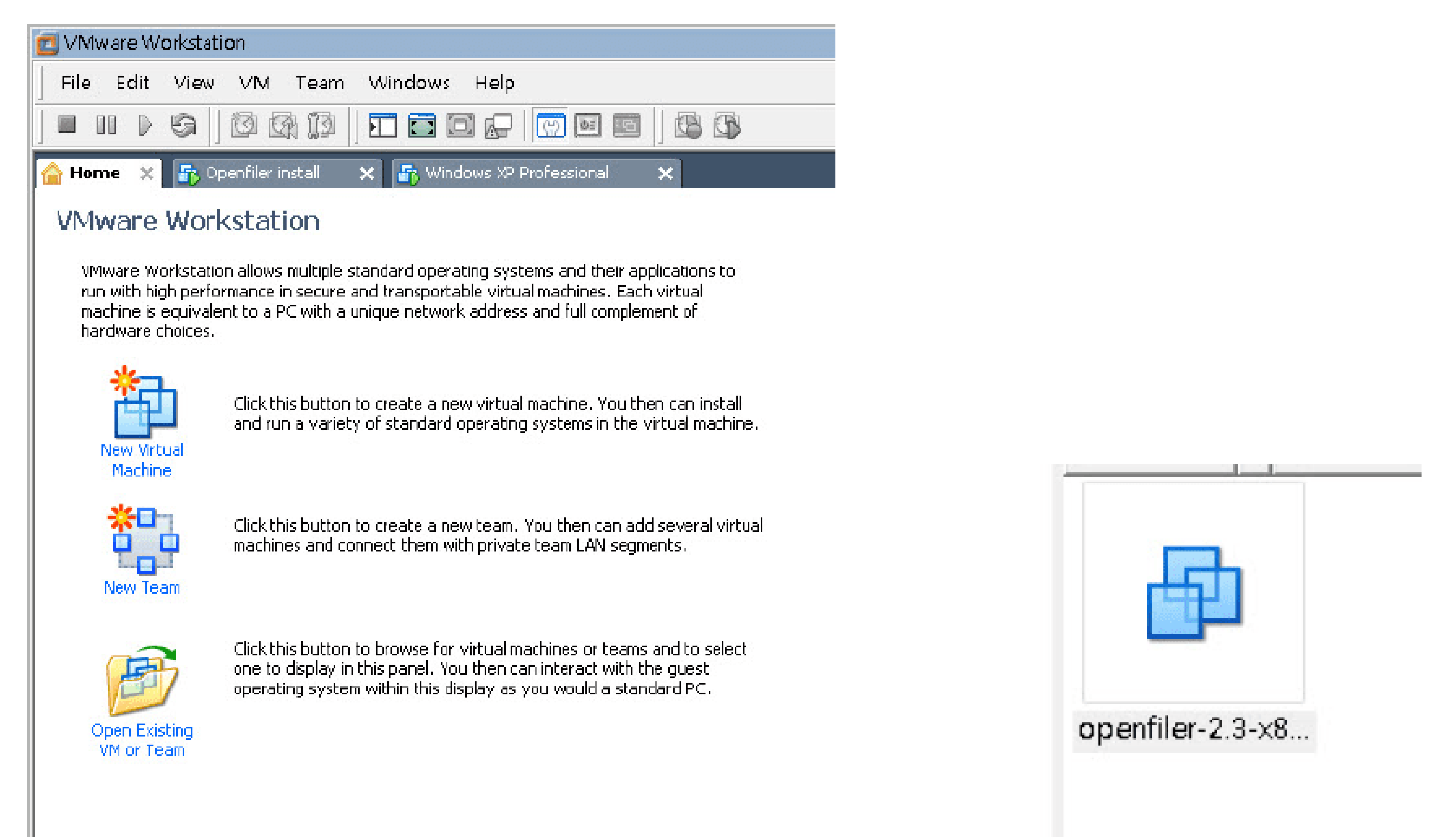

▲ 解开这个压缩文件



▲ 我们用 VMware Workstation 来运行 OpenFiler

3. 选择 Open Existing VM or Team。

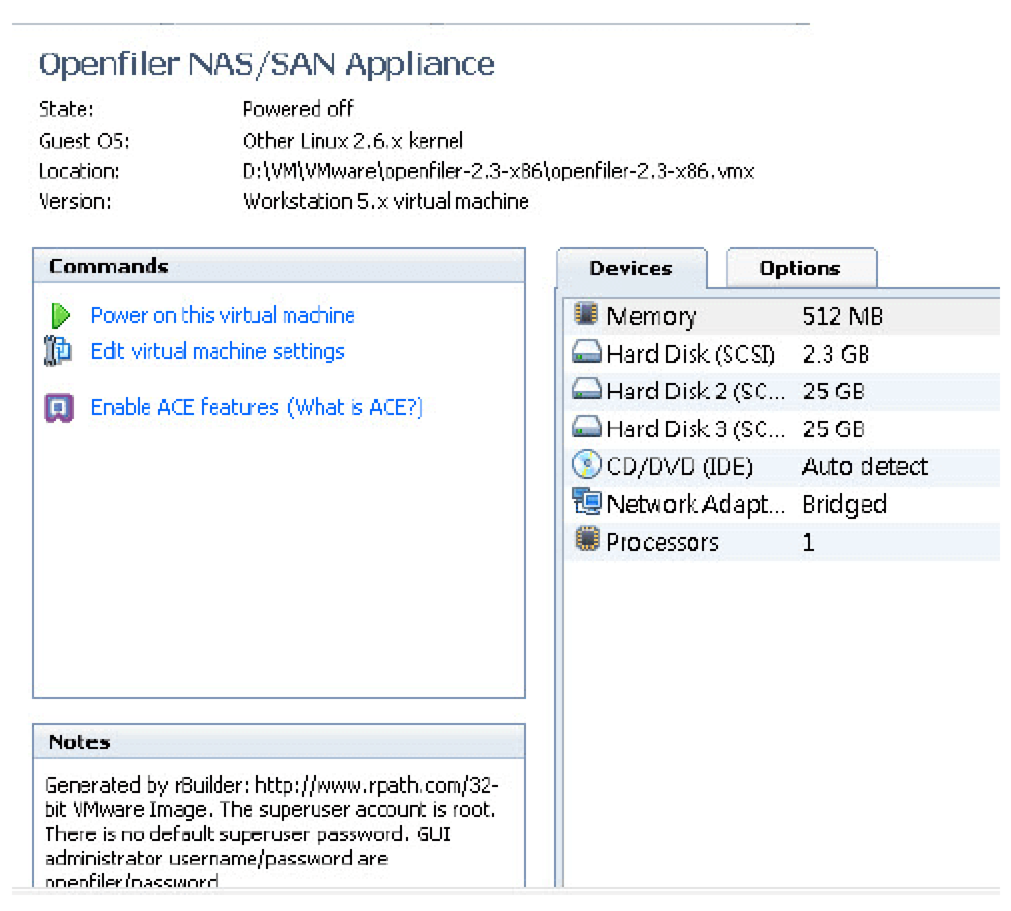
4. 选择刚才解开的 OpenFiler 文件夹中的 openfiler-2.3-x86_64.vmx。



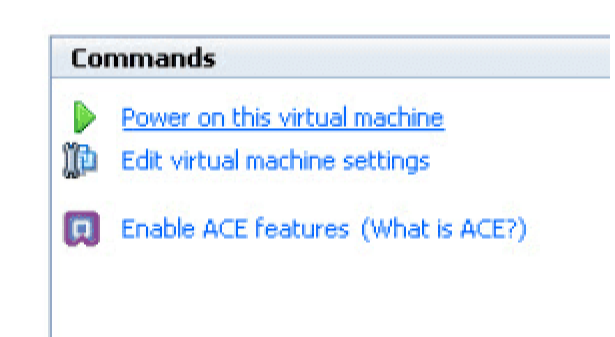
▲ 选择这里　　▲ 选择刚才解开的 VMware 格式文件

5. 开启之后，这个虚拟机其实已经可以直接使用，但这虚拟机并无任何硬盘可供分享。我们必须创建硬盘给 OpenFiler 使用。

6. 在 VMware 的主画面中，选择 Edit virtual machine settings。



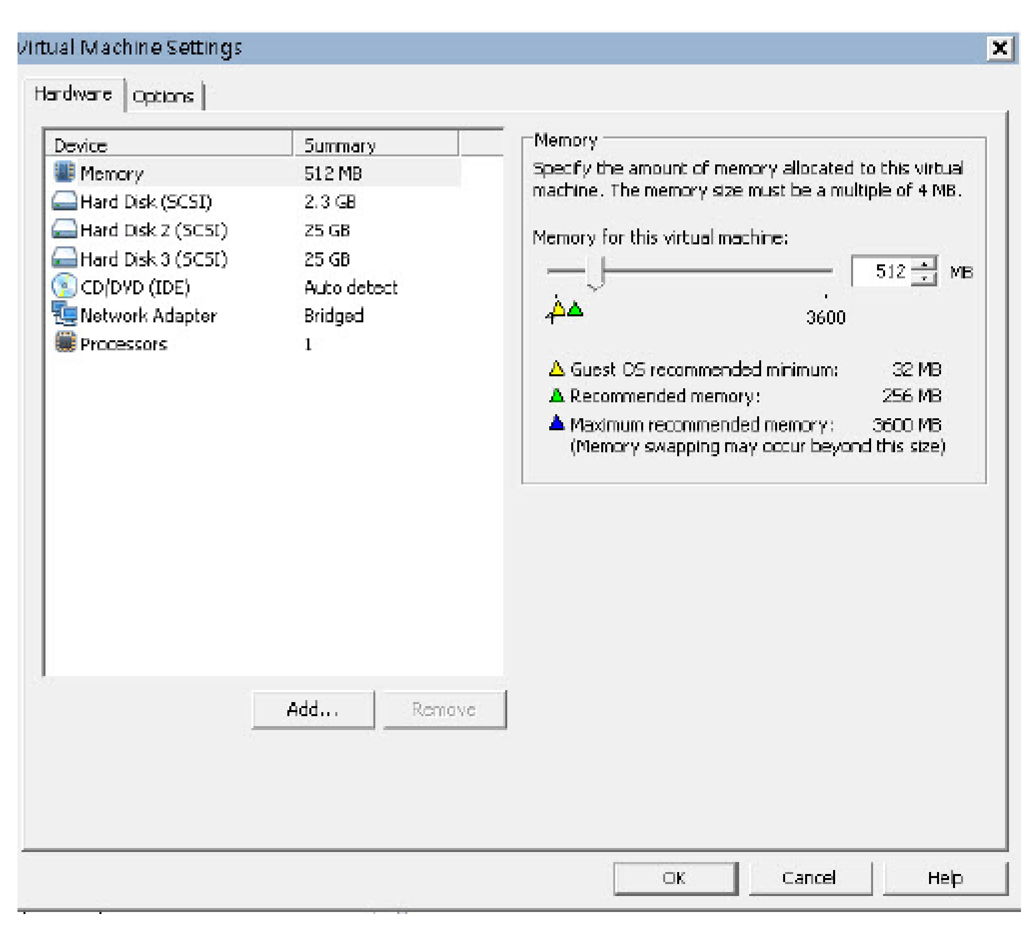




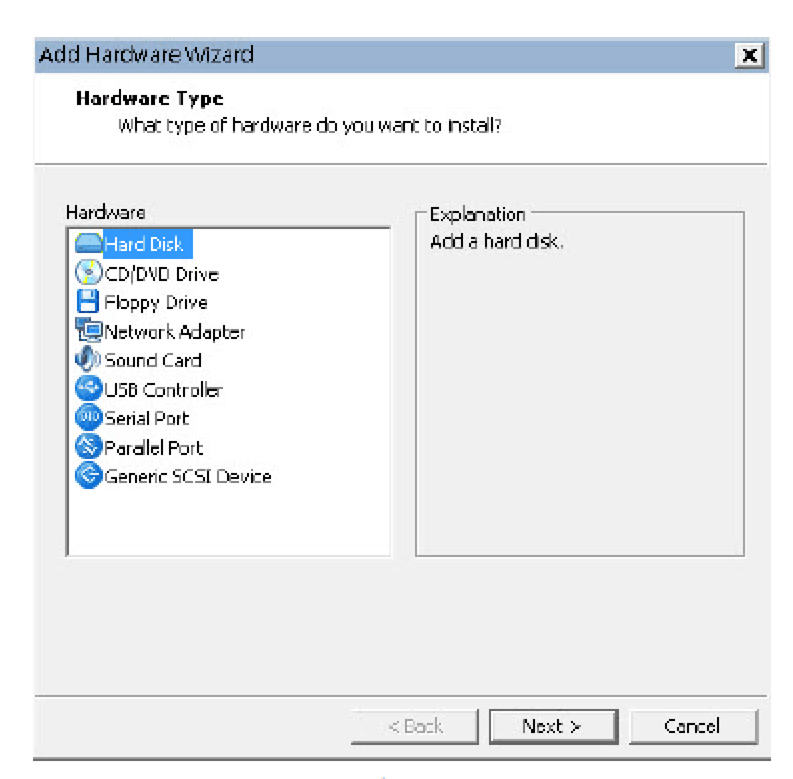
▲ 开启之后可以直接使用　　▲ 我们可以在这里新增硬设备

7. 在 Hardware 选项卡下单击 Add 按钮。如果你使用的是 Vista 或是 Windows Server 2008 以上，必须拥有系统管理员的权限才能做这个动作。

8. 进入之后选择 Hard Disk 选项，并且单击 Next 按钮。



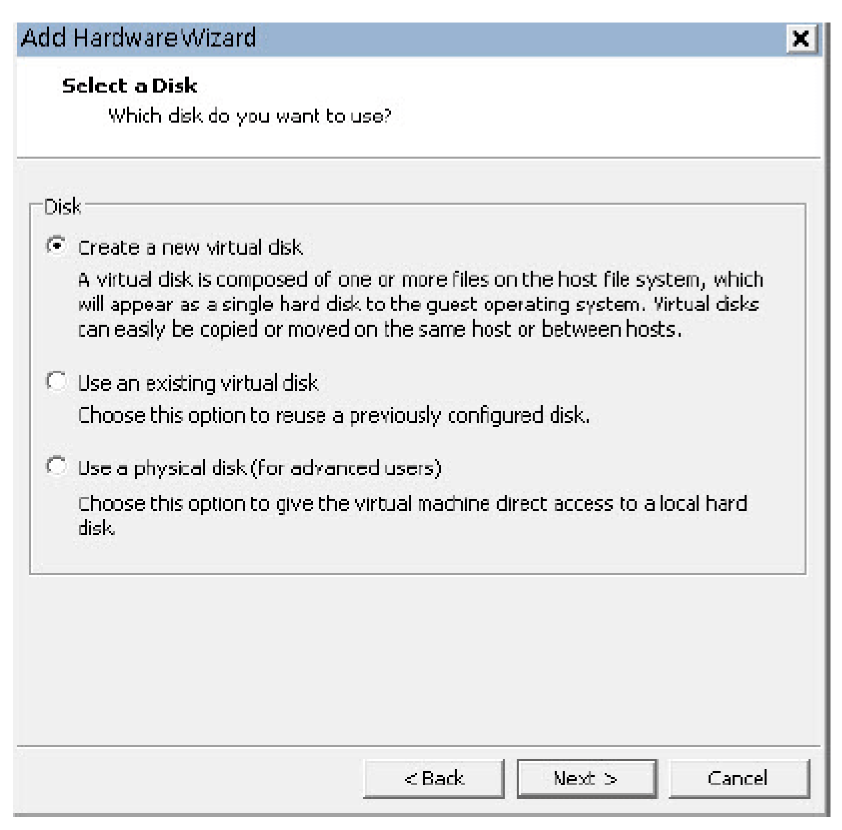
▲ 新增硬设备时需要系统管理员的权限



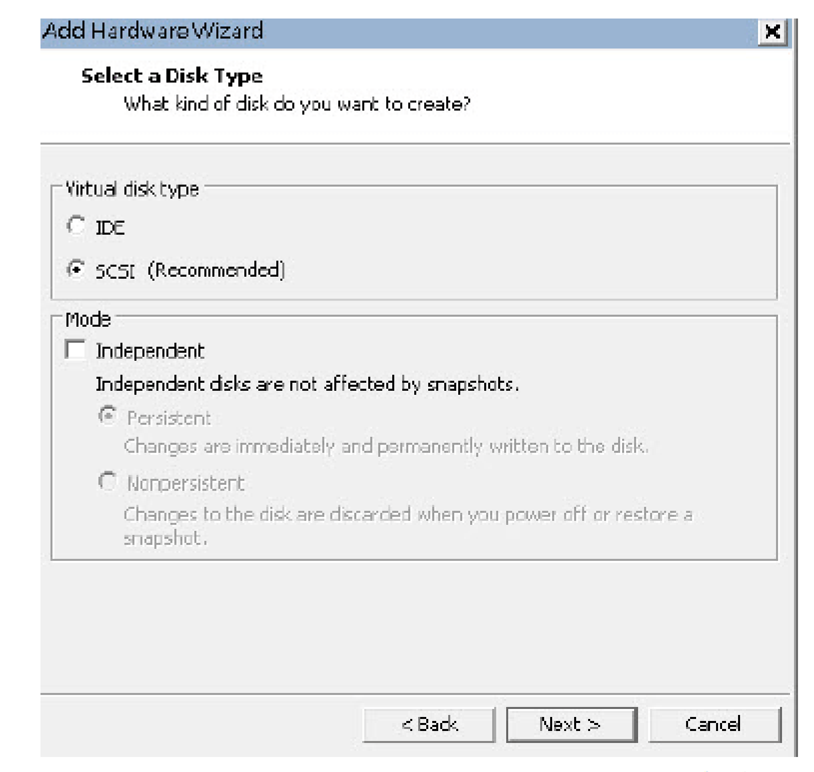
▲ 选择新增一个新的硬盘

9．选择 Create a new virtual disk 单选按钮，单击 Next 按钮。

10．在 Virtual disk type 选项卡下选择 SCSI 单选按钮，并且单击 Next 按钮。



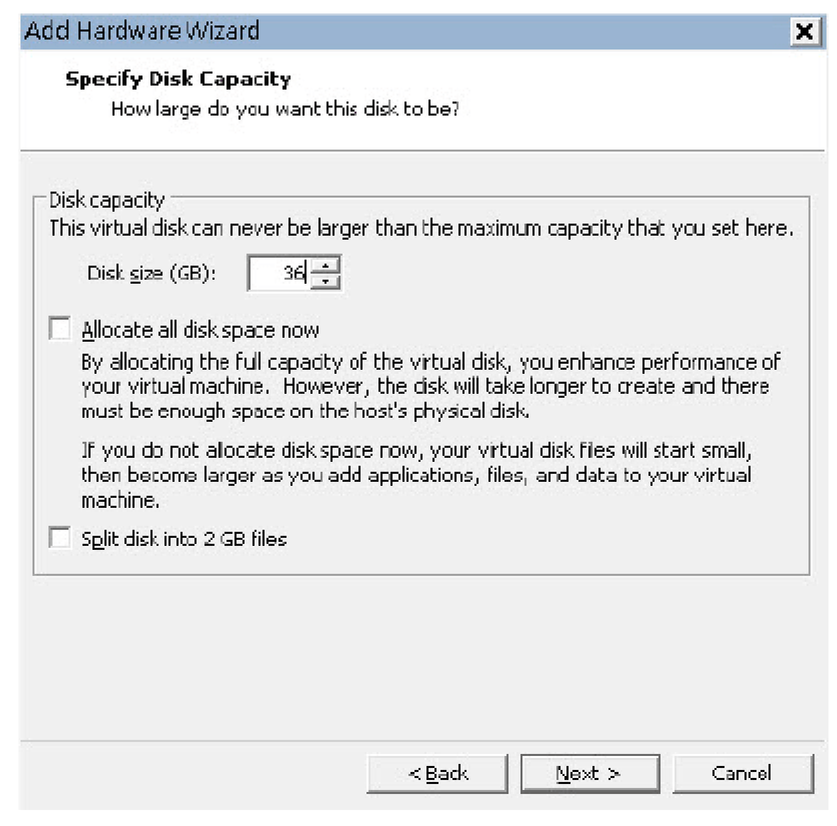
▲ 选择选项



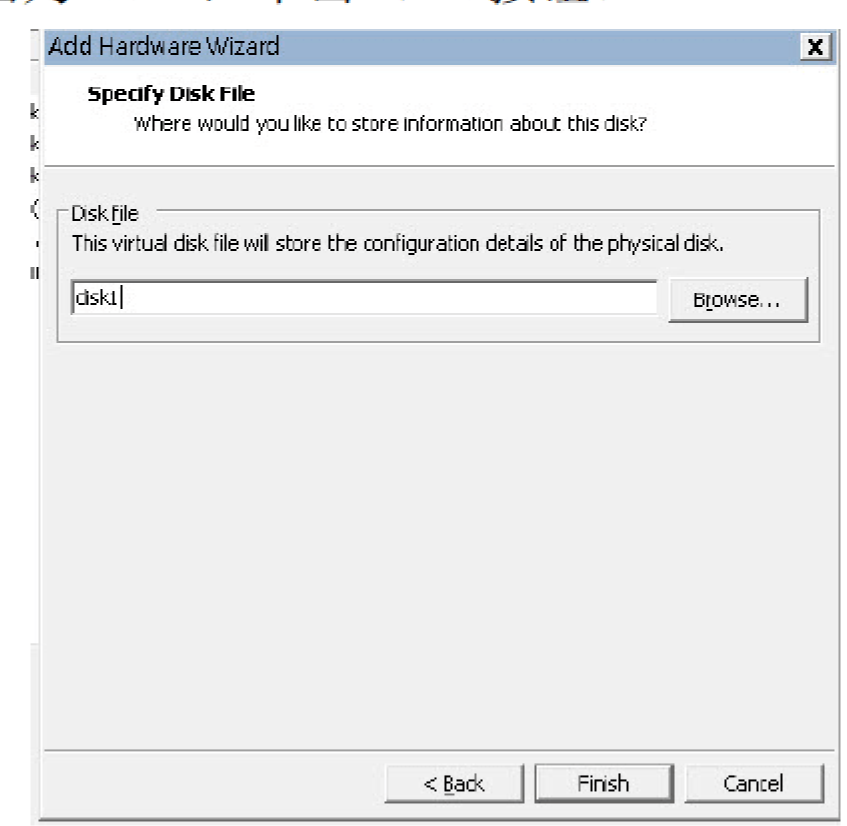
▲ 大部分的 Linux 系统对 SCSI 的硬盘兼容性较好

11．之后输入这个 SCSI 硬盘的大小，输入 50GB。此时系统并不会马上占用 25GB 的空间，除非你选中了下面的 Allocated all disk space now 单选按钮。单击 Next 按钮。

12．之后你需要将这个磁盘命名，将其命名为 disk1，单击 Next 按钮。

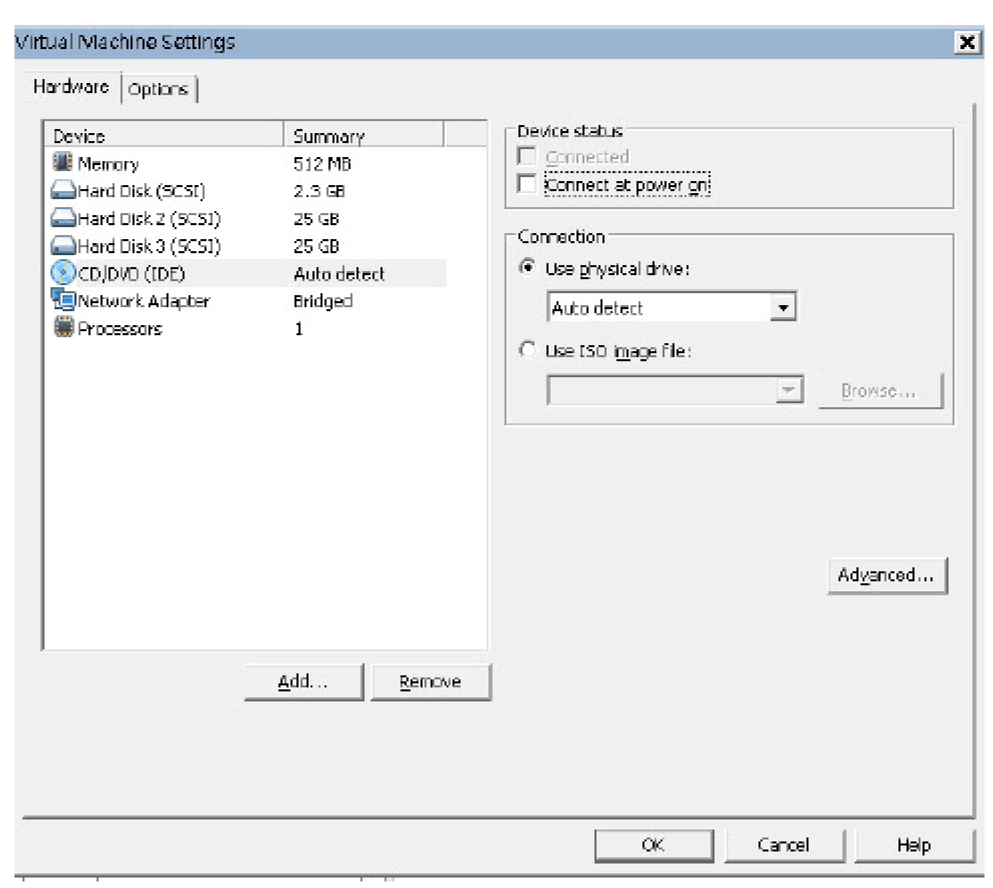


▲ 输入空间大小



▲ 替新增的磁盘命名

13．在完成自制 iSCSI 的 SAN 存储设备 OpenFiler 之后，你可以看到这个虚拟机下已经多了一个 SCSI 硬盘了。重复上述的步骤再添加一个 SCSI 硬盘。



▲ 新增完成了

14．添加完之后可以直接启动这台虚拟机。

15．当出现下图中所示的画面时，表示你在 VMware 下 OpenFiler 已经安装完毕了。我们会在配置的步骤中说明如何将新增的实体硬盘，利用 VG 中的 LVM 来制作成 LUN。



▲ 这是 OpenFiler 启动后的画面

2. 将 OpenFiler 安装到实体机器上

将 OpenFiler 安装到实体机上就和安装任何一个 Linux 操作系统一样，如果你有任何安装 Linux 的经验，安装 OpenFiler 就不会有任何问题了。要注意的是，OpenFiler 虽然是和任何正常的 Linux 一样，但其尽量简化成一个 SAN/NAS 专用的 Linux，因此许多和存储无关的功能都拿掉了，如中文语言套件和 X Window，因此我们在安装时，只能选择英文，安装时也无法选择要安装什么 Linux 套件。

▶ 安装 OpenFiler

1．首先我们将 OpenFiler 的 ISO 档烧成 CD，并且放入计算机光驱中，选择光驱开机，就会出现如下图所示的画面，直接按 Enter 键。



▲ 开始安装，就像任何一个标准的 Linux

2. 此时会出现标准的 Linux 安装画面，OpenFiler 使用的是 Fedora/RedHat 的 Anaconda 安装程序，可以看到标准的 Linux 加载画面。

```
assign_interrupt_mode Found MSI capability
assign_interrupt_mode Found MSI capability
assign_interrupt_mode Found MSI capability
assign_interrupt_mode Found MSI capability
assign_interrupt_mode Found MSI capability
pci_hotplug: PCI Hot Plug PCI Core version: 0.5
ACPI: ACPI0007:00 is registered as cooling_device0
ACPI: Processor [CPU0] (supports 8 throttling states)
Linux agpgart interface v0.103
agpgart: Detected an Intel 440BX Chipset.
agpgart: AGP aperture is 256M @ 0x0
Serial: 8250/16550 driver $Revision: 1.90 $ 4 ports, IRQ sharing enabled
serial8250: ttyS0 at I/O 0x3f8 (irq = 4) is a 16550A
serial8250: ttyS1 at I/O 0x2f8 (irq = 3) is a 16550A
00:09: ttyS0 at I/O 0x3f8 (irq = 4) is a 16550A
00:0a: ttyS1 at I/O 0x2f8 (irq = 3) is a 16550A
brd: module loaded
input: Macintosh mouse button emulation as /class/input/input0
Uniform Multi-Platform E-IDE driver
ide: Assuming 33MHz system bus speed for PIO modes; override with idebus=xx
PIIX4: IDE controller (0x8086:0x7111 rev 0x01) at  PCI slot 0000:00:07.1
PIIX4: not 100% native mode: will probe irqs later
PIIX4: IDE port disabled
    ide0: BM-DMA at 0x10c8-0x10cf
_
```

▲ 加载标准的 Linux 设备清单

3. 如下图所示不需要检查光盘，单击 Skip 按钮。



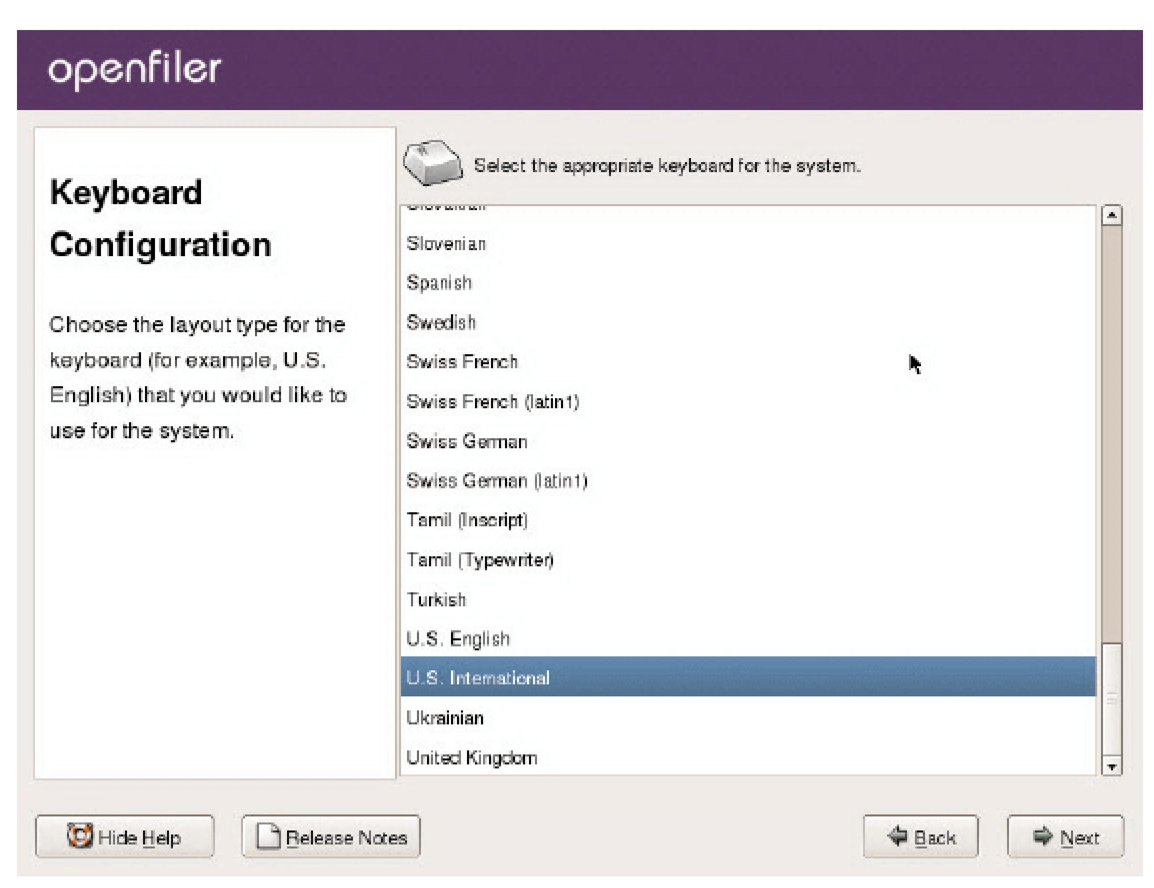
▲ 不需检查光盘

4. 此时会出现标准的安装画面，单击 Next 按钮。



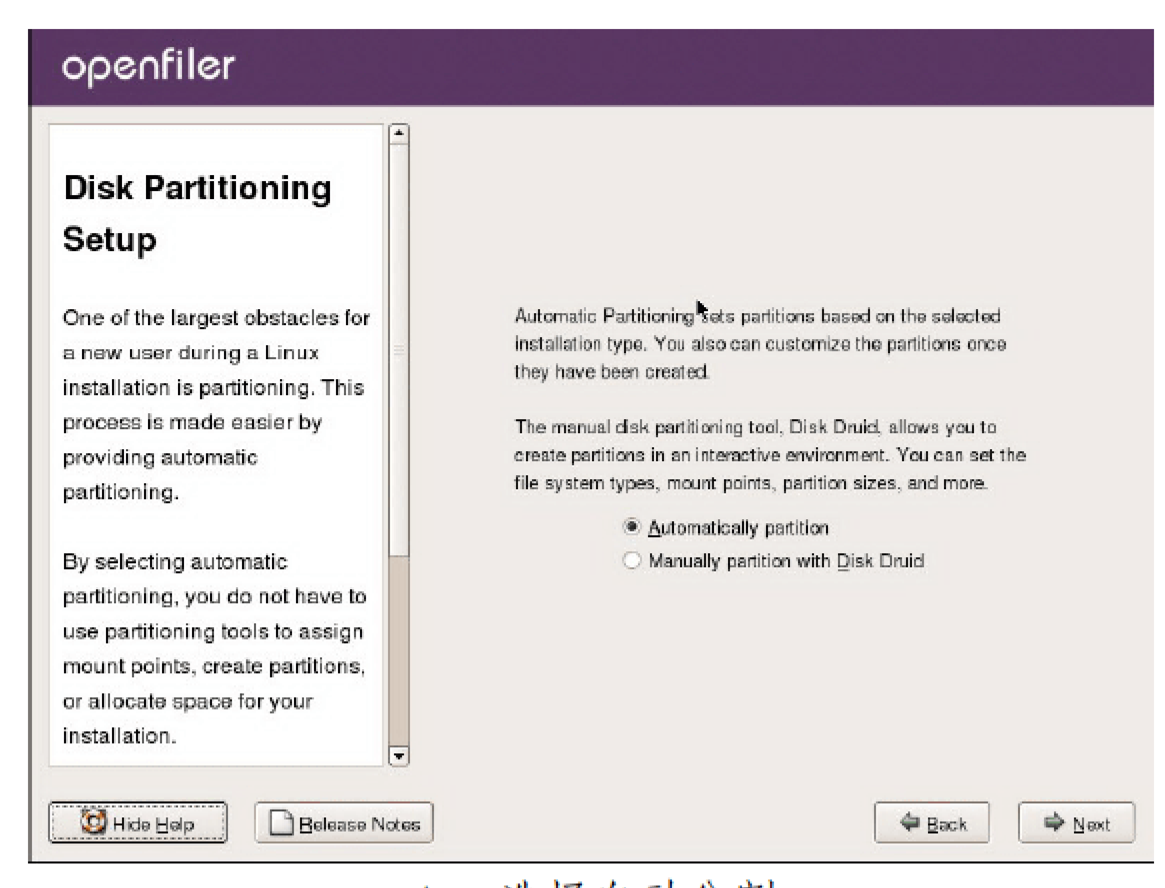
▲ 标准安装画面，直接单击 Next 按钮

5. 选择语系，只有西语系我们就选择 US English，之后单击 Next 按钮。



▲ 选择英文安装

6. 接下来是安装磁盘分区。要注意的是，这里的磁盘分区是安装 OpenFiler 的主程序，和之后给虚拟机使用的独立存储是完全不同的，我们可以直接使用 OpenFiler 预设的分割方式。选择 Automatically Partition 即可。单击 Next 按钮。



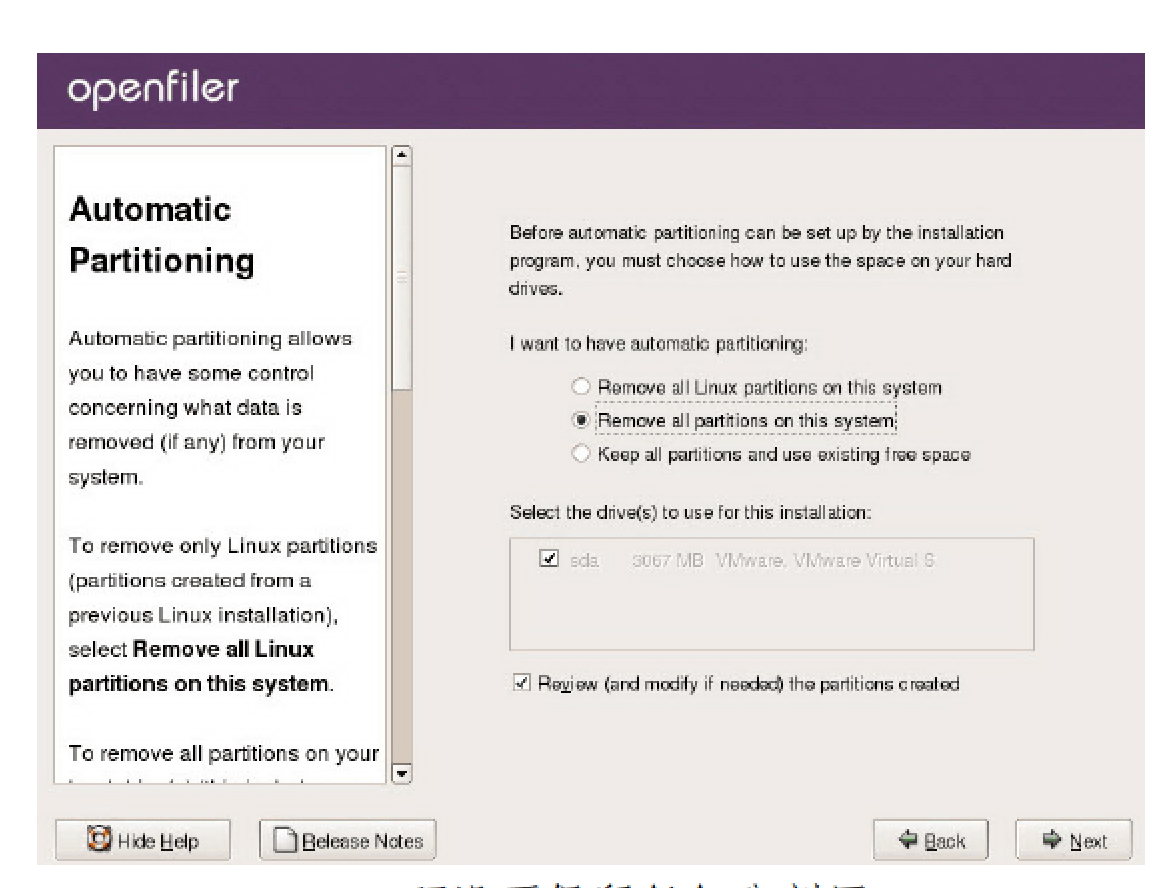
▲ 选择自动分割

7．在选择自动分割之后，磁盘上所有的数据都会不见，单击 Yes 按钮。



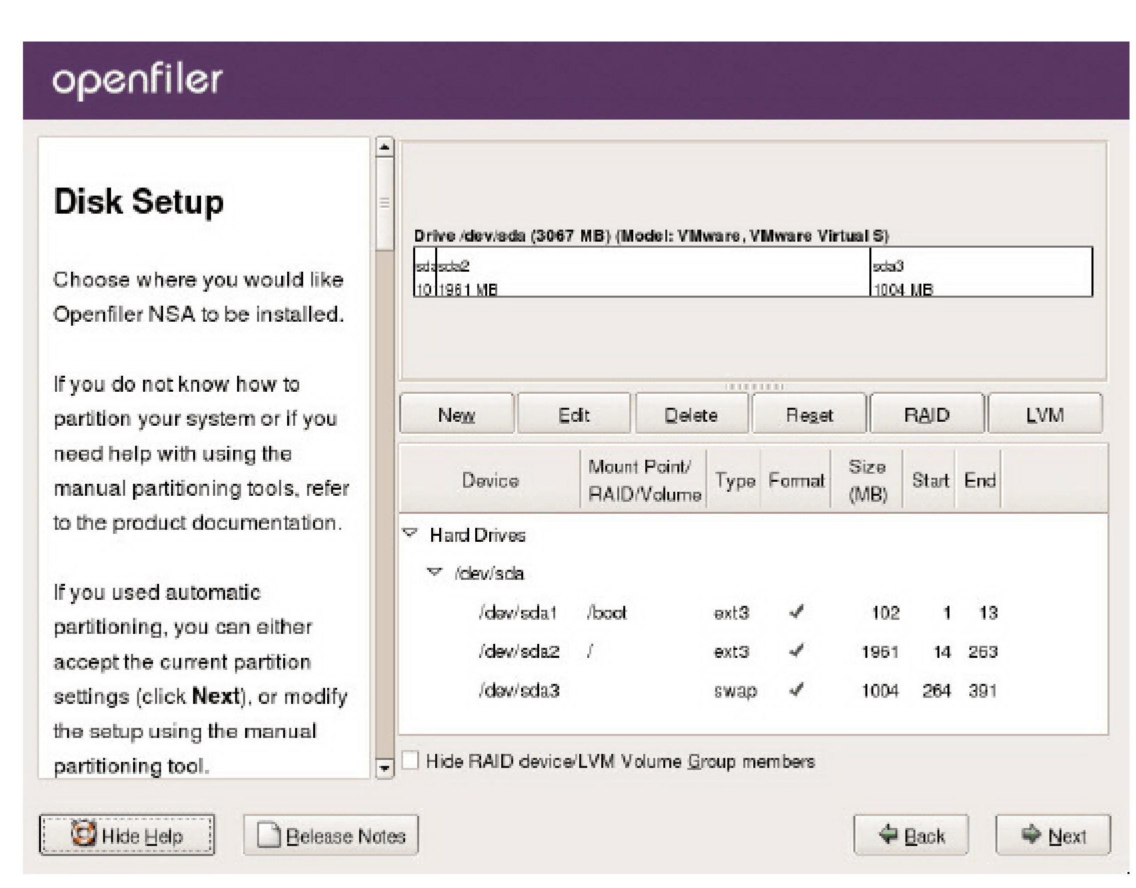
▲ 默认重新配置硬盘，所有的数据都会不见

8．我们假设你安装在全新的硬盘上，因此将所有该硬盘上的分割区都删除，单击 Yes 按钮。



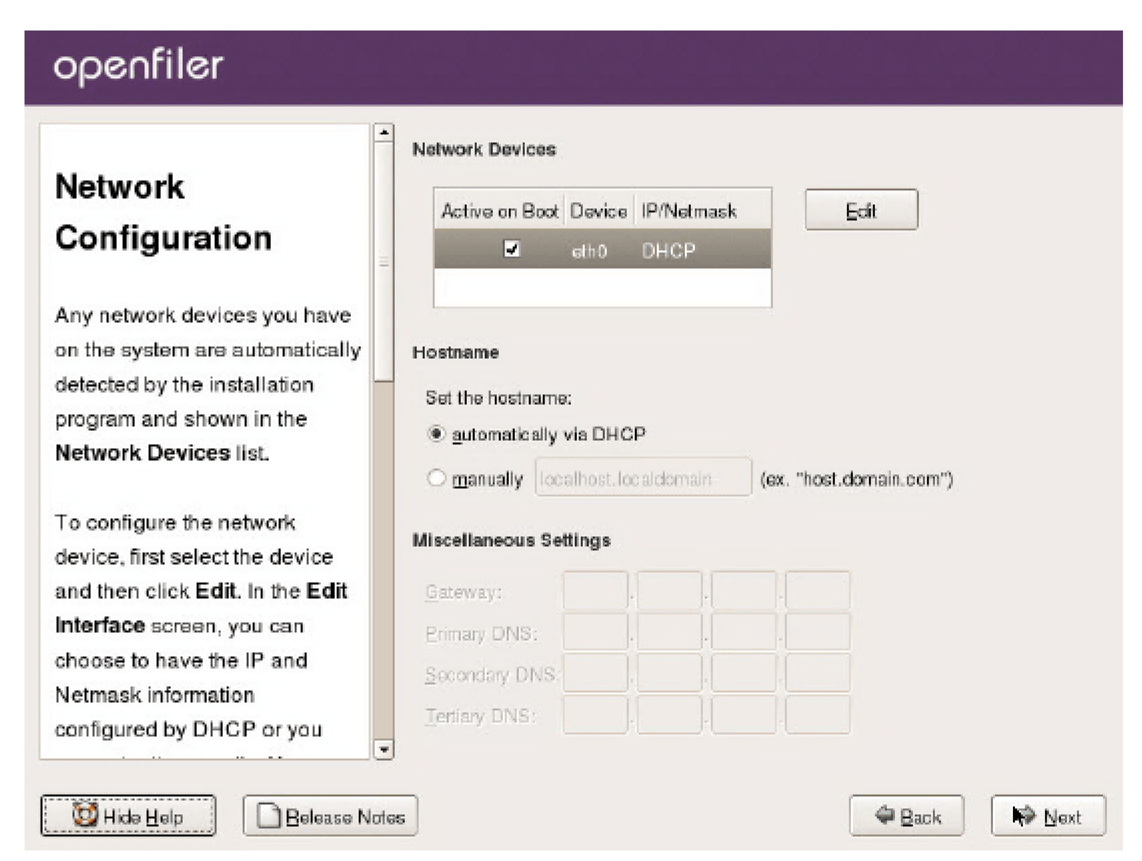
▲ 预设不保留任何分割区

9．之后 OpenFiler 会依其内定值分割硬盘，检视完毕直接单击 Next 按钮即可。



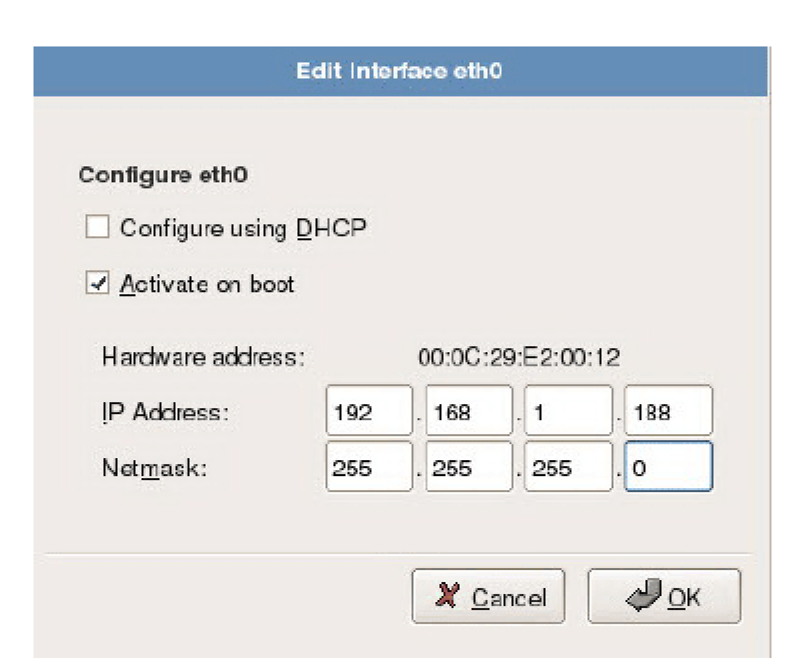
▲ OpenFiler 会照预定的比例来分割硬盘

10．接下来是配置网络，我们建议直接使用手动配置，不要用 DHCP，除非你能在路由器中配置静态 DHCP。单击 Edit 按钮。



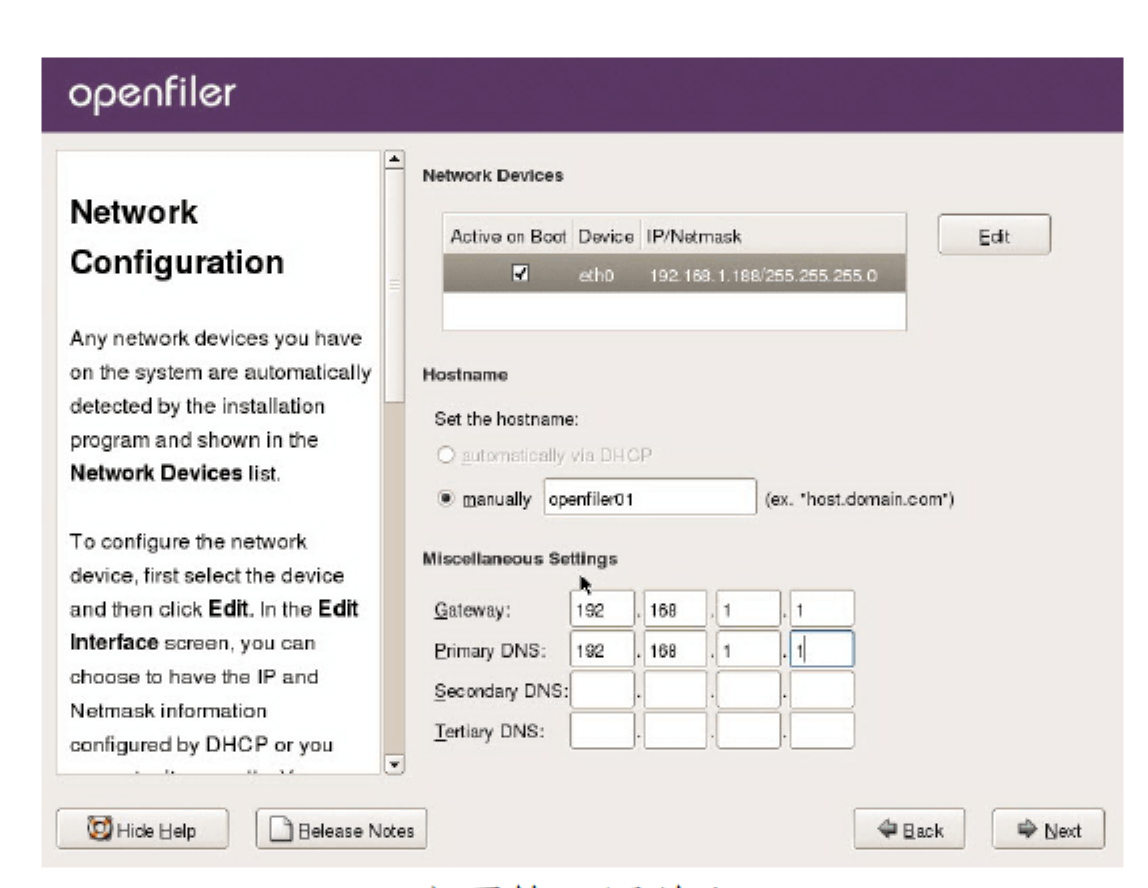
▲ 不使用 DHCP，手动编辑 IP

11．此时会进入 IP 配置的方块，输入 IP 及子网掩码，完毕后单击 OK 按钮。



▲ 输入基本内网的 IP 段，当然 OpenFiler 也可以走外网 IP

12．此时会回到上一个画面，别忘了输入网关和 DNS 之类的配置值。输入完毕单击 Next 按钮。



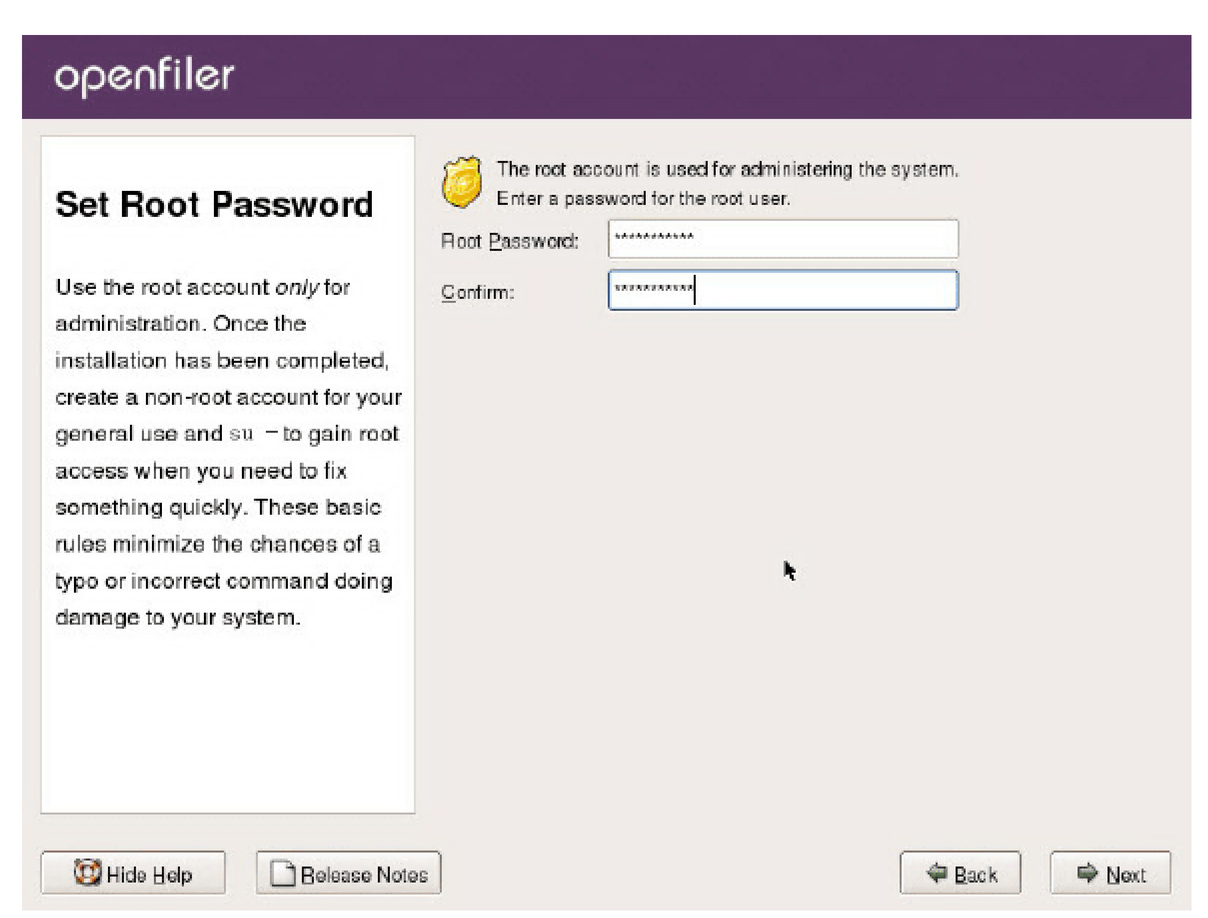
▲ 也要输入网关和 DNS

13．接下来选择时区选择所在的地区，并单击 Next 按钮。



▲ 选择时区

14．接下来是配置密码，注意这里是 root 的密码。在配置密码后，你可以在 OpenFiler 的机器用 root 账号和这个密码登录，也可以在 Web 的接口登录。



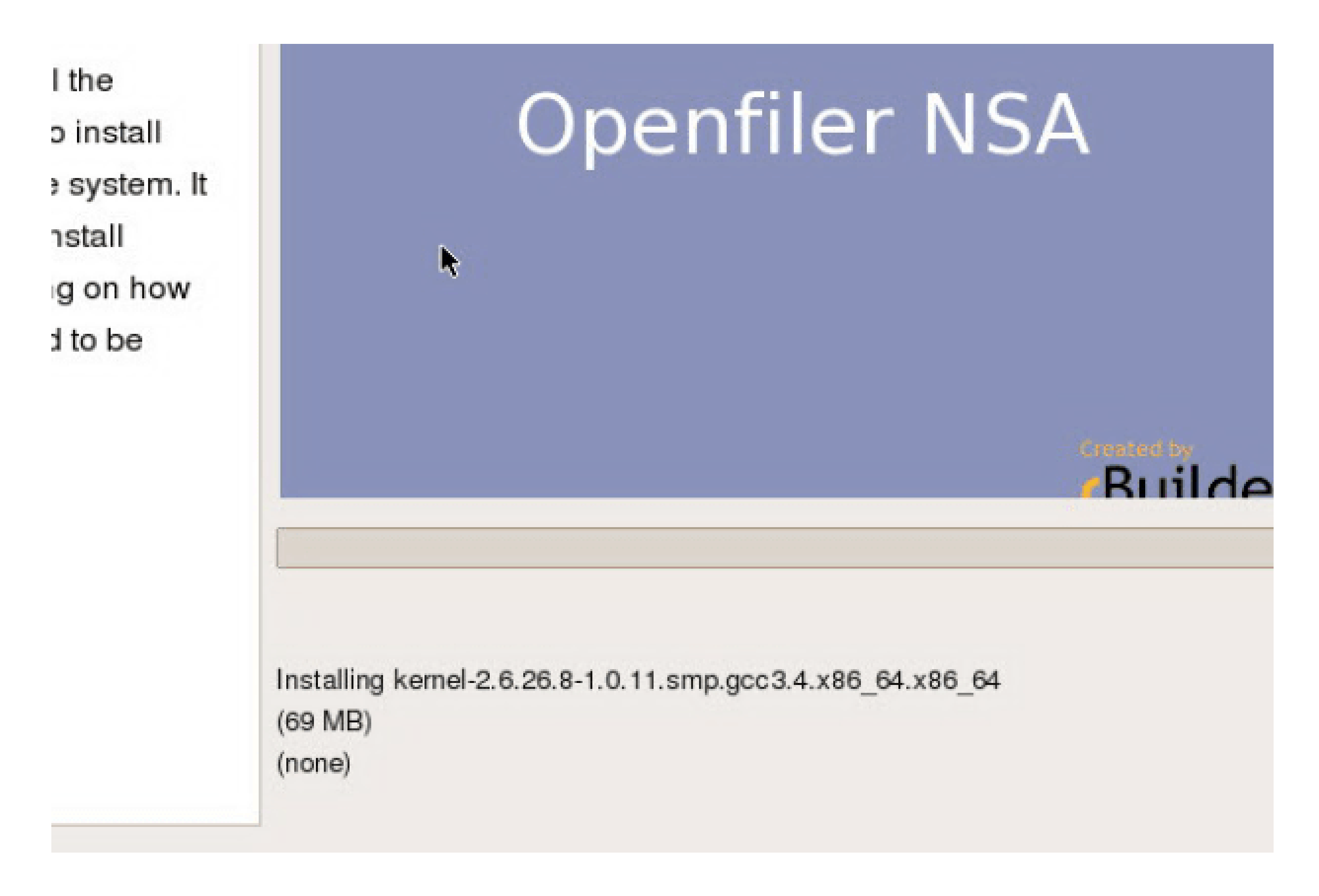
▲ 配置 root 账号的密码

15．之后就是准备安装的画面了，直接单击 Next 按钮。



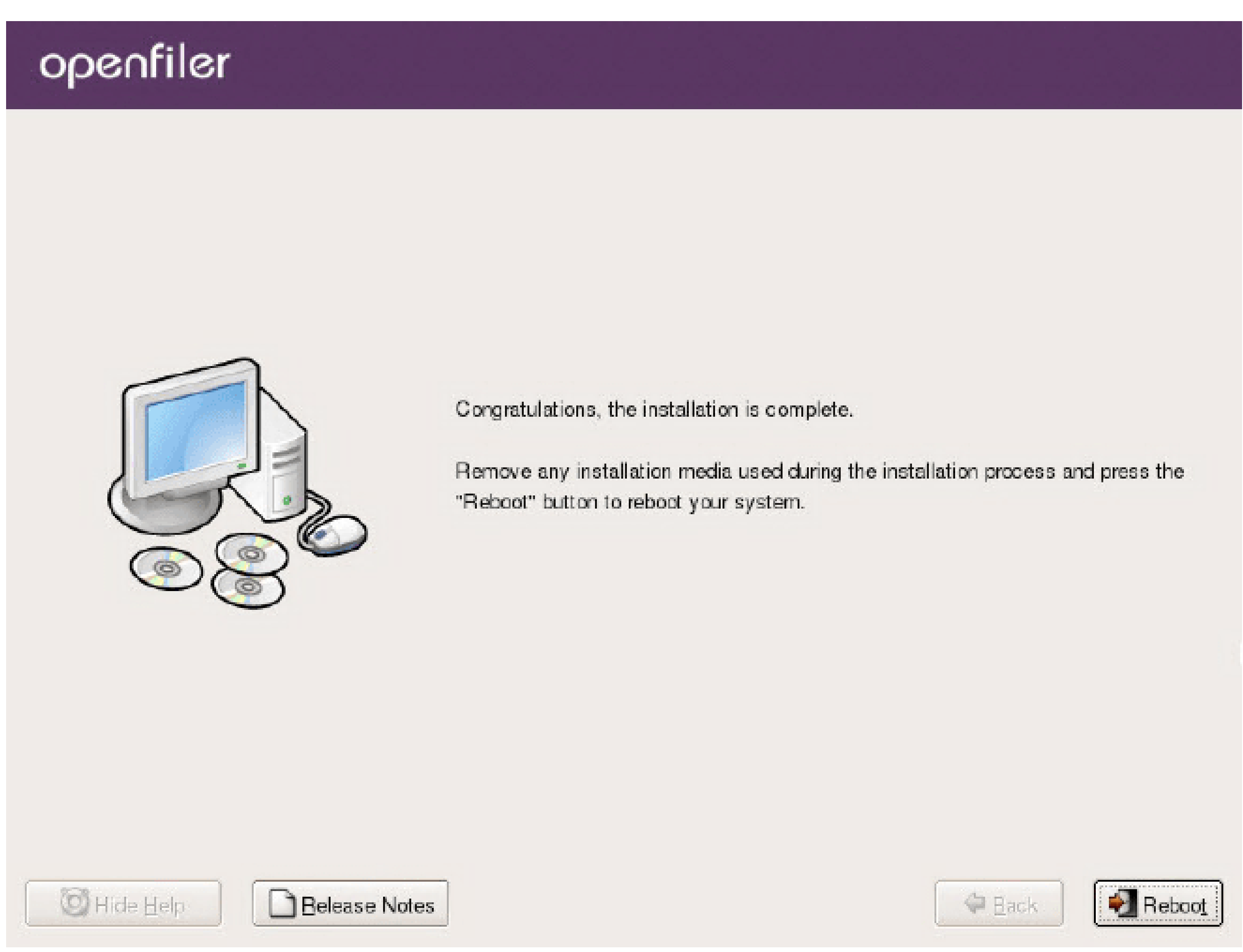
▲ 准备开始安装了

16．开始安装，大约会花 10 分钟将套件安装至硬盘中。



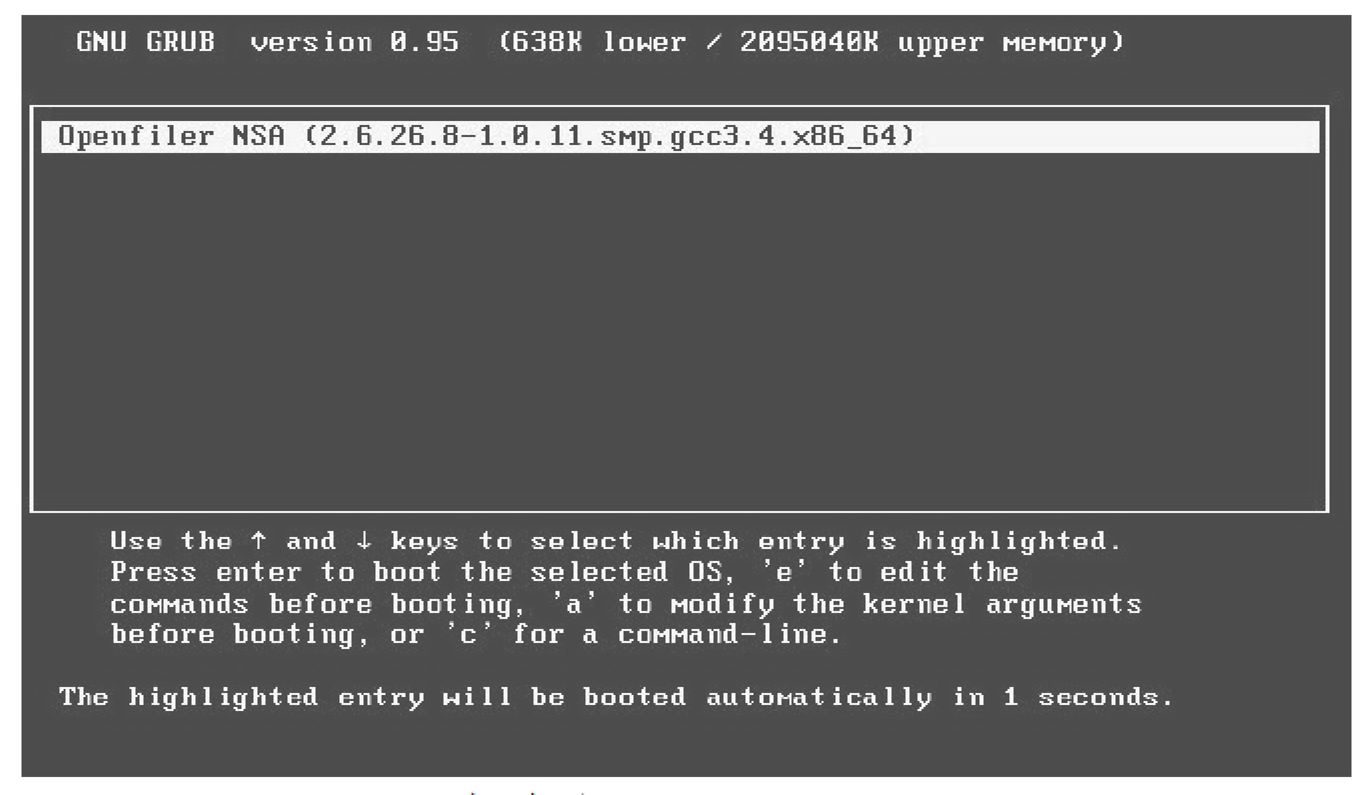
▲ 开始安装套件至硬盘中

17．当出现如下图所示的画面时，表示安装完毕，直接单击 Reboot 按钮即重新启动。



▲ 安装完毕了

18．当开机时，会出现标准的 Linux GRUB 画面，直接按 Enter 键开机。



▲ 标准的 RedHat GRUB 画面

19．开机后会加载 OpenFiler 的驱动，就像一个 Linux 一样。

```
root (hd0,0)
 Filesystem type is ext2fs, partition type 0x83
kernel /vmlinuz-2.6.26.8-1.0.11.smp.gcc3.4.x86_64 ro root=LABEL=/  quiet
   [Linux-bzImage, setup=0x3000, size=0x219300]
initrd /initrd-2.6.26.8-1.0.11.smp.gcc3.4.x86_64.img
   [Linux-initrd @ 0x37e4c000, 0x1a30be bytes]


Decompressing Linux... Parsing ELF... done.
Booting the kernel.


ide_generic: I/O resource 0x1F0-0x1F7 not free.
ide_generic: I/O resource 0x170-0x177 not free.
Red Hat nash version 4.2.15 starting
sd 0:0:0:0: [sda] Assuming drive cache: write through
sd 0:0:0:0: [sda] Assuming drive cache: write through
INIT: version 2.85 booting
Setting default font (latarcyrheb-sun16):                  [  OK  ]

                Welcome to Openfiler NSA
                Press 'I' to enter interactive startup.
Starting udev:                                             [  OK  ]
Initializing hardware...  storage network audio_
kernel direct mapping tables up to 100000000 @ 8000-d000
```

▲　加载 Linux 驱动程序

20．出现如下图所示的画面时，表示 OpenFiler 已经安装完毕了。

```
_/_/   _/_/_/     _/_/_/  _/    _/    _/      _/  _/   _/_/_/  _/
      _/
     _/

------------------------------------------------------------------------
          Commercial Support: http://www.openfiler.com/support/         |
  Administrator Guide: http://www.openfiler.com/buy/administrator-guide |
    Community Support: http://www.openfiler.com/community/forums/       |
   Internet Relay Chat: server: irc.freenode.net    channel: #openfiler |
------------------------------------------------------------------------
           (C) 2001-2008 Openfiler. All Rights Reserved.                |
    Openfiler is licensed under the terms of the GNU GPL, version 2     |
               http://www.gnu.org/licenses/gpl-2.0.html                 |
------------------------------------------------------------------------


elcome to Openfiler NSA, version 2.3

eb administration GUI: https://192.168.1.188:446/

penfiler01 login: _
```

▲　安装完毕，可以使用了

12.2　使用 OpenFiler 基本功能

OpenFiler 可以说是一个相当完整的 Linux 系统，因此使用上十分复杂，但我们只需要针对虚拟机的存储方面进行配置即可。在 OpenFiler 上，最常用的当然就是针对本身的使用（如开关机、更换 IP 等），针对硬盘的分割配置（VG，LUN 等）、NAS 的分享配置以及最重要 iSCSI 的创建及连接。

12.2.1　了解基本操作

OpenFiler 是标准的 Linux，并且是以 RedHat 为基础，因此在操作上和 RedHat 大同小异。在本机 OS Console 使用上，主要都是针对 Linux 系统本身的操作，而大部分的存储设备操作，则都在 Web 接口上。首先我们必须先用 root 账号登录到 OpenFiler 的操作系统，或是使用 SSH 从远程登录，下面就是几个常见的操作。

1. OpenFiler 的基本使用

最常用的 OpenFiler 操作还是在 Linux 本身，如果要配置存储方面的操作就要使用 Web 接口了。

▶ 将 OpenFiler 关机

1. 进入 OpenFiler 的终端机接口。

```
[root@openfiler01 ~]#
[root@openfiler01 ~]# ls  -l
total 308
-rw-------  1 root root   1209 Feb  7 19:01 anaconda-ks.cfg
-rw-r--r--  1 root root 147093 Feb  7 19:00 conary-tag-script
-rwxr-xr-x  1 root root  68880 Feb  7 19:00 conary-tag-script.output
-rw-r--r--  1 root root  73781 Feb  7 19:01 install.log
-rw-r--r--  1 root root    971 Feb  7 19:01 install.log.syslog
[root@openfiler01 ~]# _
```

▲ 进入终端机接口

2. 输入 shutdown –h now，系统即进入关机模式。

```
[root@iscsi ~]#
[root@iscsi ~]#
[root@iscsi ~]# shutdown -h now

Broadcast message from root (tty1) (Sun Feb  8 14:02:18 2009):

The system is going down for system halt NOW!
INIT: Switching to runlevel: 0
INIT: Sending processes the TERM signal
_
```

▲ 输入关机指令后，系统就开始关机了

2. 更改 OpenFiler 的 root 密码

如果你使用的是虚拟机 OpenFiler，其终端机登录的 root 是没有密码的。我们可以在这里替 root 加上密码。

▶ 替 OpenFiler 的 root 加上密码

1. 首先以 root 登录，当系统询问密码时，直接按 Enter 键进入。
2. 在终端机画面下输入 passwd，按 Enter 键。
3. 此时系统会要求输入两次密码。



▲ 输入 passwd 指令



▲ 输入密码

4. 当出现如下图所示的提示时，表示密码配置完毕，下次登录就要用这个密码登录，而在 Web 接口上，也可以使用 root 和新密码登录配置 SAN。

```
[root@localhost /]#
[root@localhost /]# passwd
Changing password for user root.
New UNIX password:
Retype new UNIX password: _
```

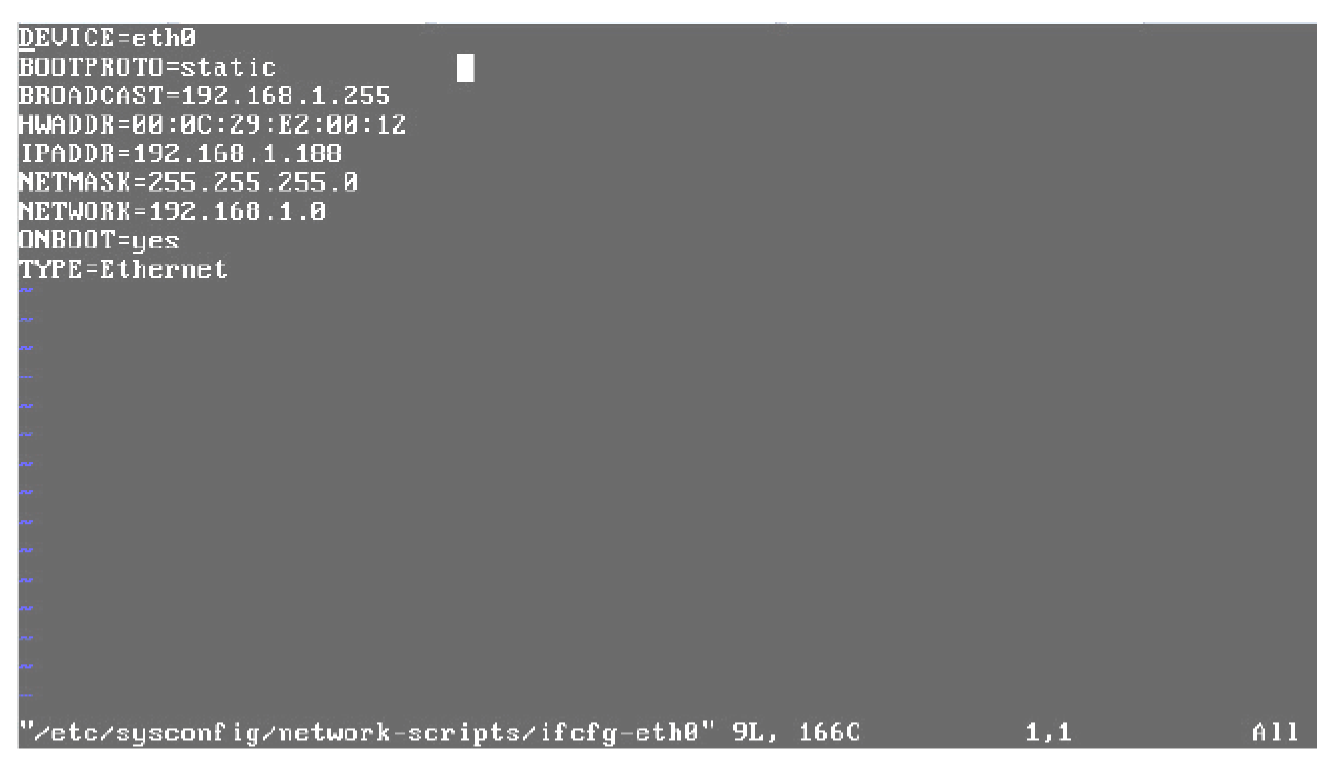
▲ 密码输入完成

3. 更换 OpenFiler 的网络配置值

如果你想更换 OpenFiler 的 IP，而不是使用动态分配的 DHCP，也是使用标准 Linux 的方法，下面就是详细的步骤。

▶ 更换 OpenFiler 的 IP、DNS 及主机名

1. 以 root 身份进入 OpenFiler 的终端机方块，并且使用 vi 编辑 IP 配置文件，输入 vi /etc/sysconfig/network-scripts/ifcfg-eth0，前提是你的网卡使用的是 eth0。



▲ 编辑这个文件

2. 此时会进入配置文件的编辑，下面就是要修改的地方：

- DEVICE=eth0：这是网卡代号，必须要与 ifcfg-eth0 相对应。
- BOOTPROTO=static：开机协议，有 dhcp 及 static （none），改成 static。
- BROADCAST=192.1612.1.255：广播地址，这个一般不用动。
- HWADDR=00:40:D0:13:C3:46：网卡 MAC，这也不需要动。
- IPADDR=192.1612.1.13：这里就是你要改成的 IP 值。
- NETMASK=255.255.255.0 ：子屏蔽网络。
- NETWORK=192.1612.1.0 ：网域第一个 IP。
- GATEWAY=192.1612.1.1 ：默认网关。
- ONBOOT=yes ：是否开机启动。

3. 修改完之后，先将这个文件存盘，然后离开 vi。

4. 之后重新启动这片网卡，先输入 ifdown eth0，然后再输入 ifup eth0，这片网卡就重新启动了，以后 OpenFiler 也使用这个 IP 了。

5. 如果要变更 DNS，可以修改 /etc/resolv.conf 这个文件，输入 nameserver XXX.XXX.XXX.XXX 的 IP 地址即可。通常内网的 DNS 会放在第一个。



▲ 编辑这个文件来更改 DNS

6. 使用免安装版的 OpenFiler 时，主机名为 localhost，如果你有自己架设 DNS，并且希望将这台 OpenFiler 可以用主机名找到，可以修改这台 OpenFiler 的主机名，只要在终端机方块下输入“hostname <新的主机名>”，如 hostname openfiler02 即可。

```
[root@openfiler01 /]#
[root@openfiler01 /]#
[root@openfiler01 /]# hostname openfiler01
[root@openfiler01 /]# hostname
openfiler01
[root@openfiler01 /]# _
```

▲ 直接用 hostname 指令来更改主机名

12.2.2 实体硬盘到逻辑硬盘（LUN）的操作：使用 OpenFiler

当你安装好 OpenFiler 之后，接下来就是要将 OpenFiler 下的磁盘分享出来给虚拟机或网络上的其他主机使用了。在标准的 SAN 之后，这些可以在 RAID 层面完成，但 VG 的好处及弹性是 RAID 无法比较的，因此我们就假定有两个 RAID 1 的实体硬盘，看这两个硬盘在 OpenFiler 下的 VG 是如何一步一步创建的。

▶ 创建 VG 的步骤

（1）进入 OpenFiler 的接口，并且选择要使用的实体硬盘。

（2）将要加入的实体硬盘格式化成 Physical Volume 格式。

（3）创建一个 VG 组，并且将格式化成为 PV 格式的实体硬盘加入。

（4）加入完毕之后，就成为一个大的 VG 组，被视为系统的一个大实体硬盘。

（5）在这个 VG 中添加逻辑分割区 LUN，在 OpenFiler 中称为 Logical Volume。

（6）指定 LUN 的文件格式，如 iSCSI、ext3 或是 NFS，并且格式化。

（7）如果是 iSCSI 则需要再配置，如果是其他文件格式，就可以用 NAS 的方式分享出去而非 SAN 的方式。

1. 熟悉 OpenFiler 的 Web 管理接口

OpenFiler 大部分的存储功能都是在 Web 管理接口中实现，我们就先从熟悉其界面开始。

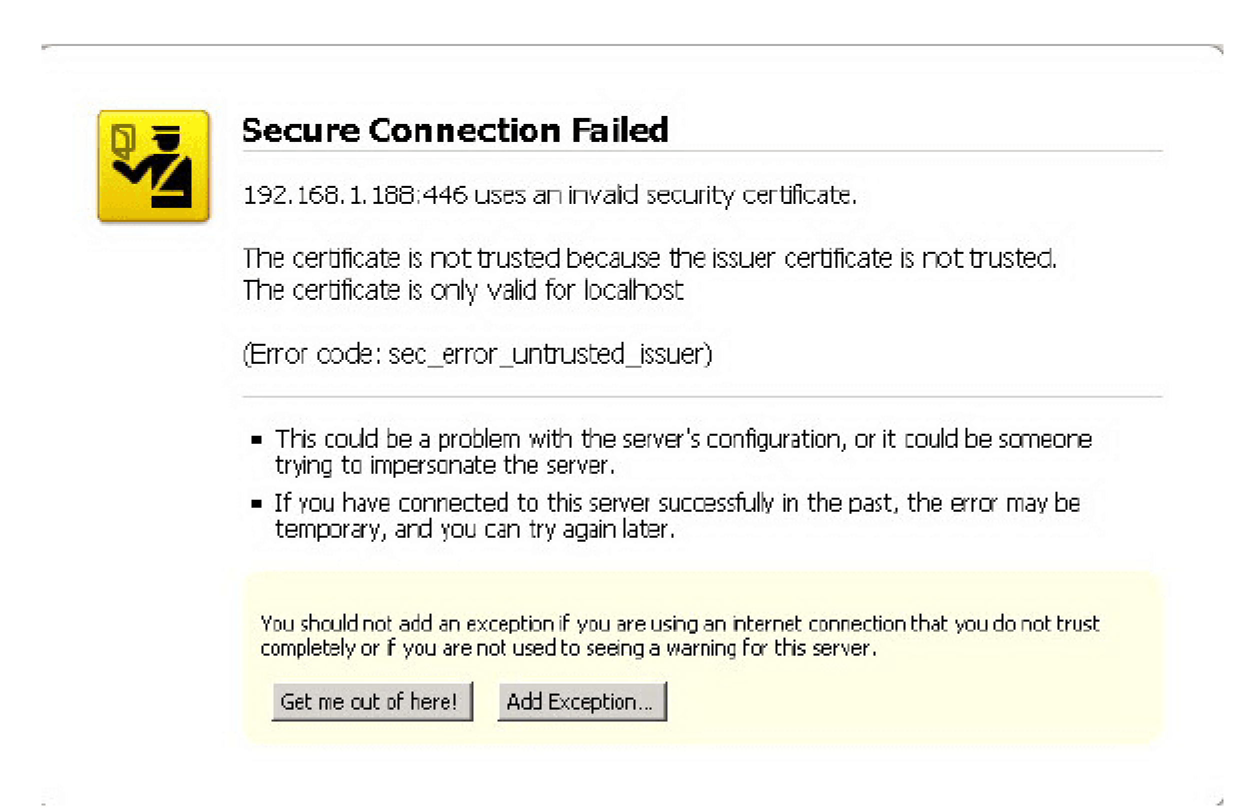
▶ 使用 OpenFiler 的接口

1．如果是使用 Firefox，则先进入浏览器，在网址列上输入 https://<OpenFiler 的 IP 地址>:446，如https://192.1612.1.188:446。



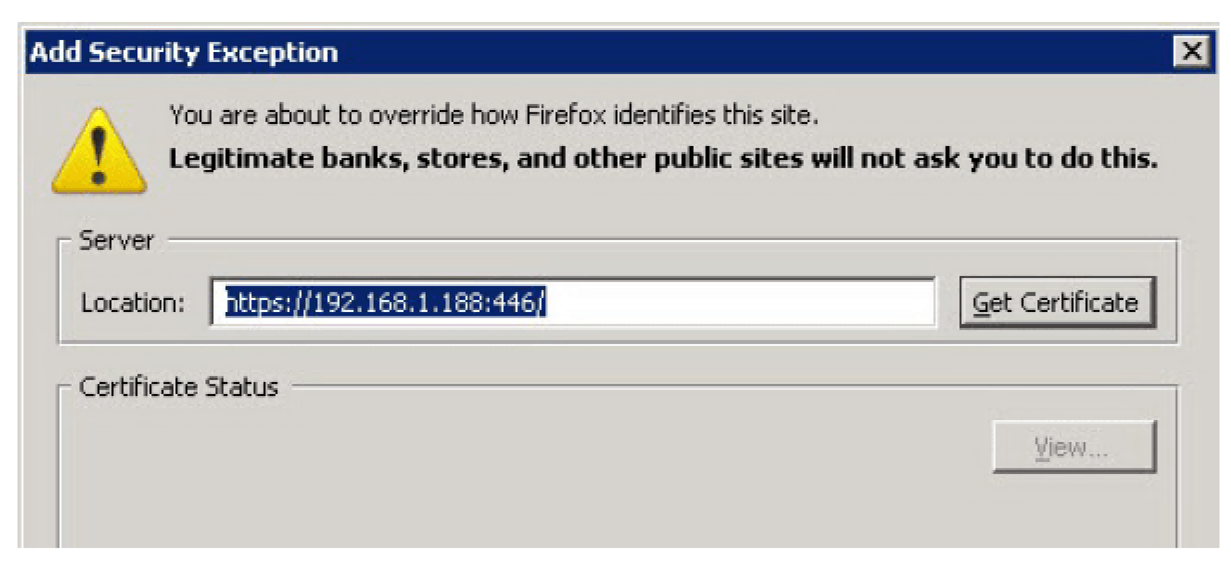
▲ 输入网址

2．此时由于这不是安全认证下的网站，因此系统会先出现如下图所示的错误信息，我们必须将这个网站加入可信任的网站，单击 or you can add an exception。此时单击 Add Exception 按钮。

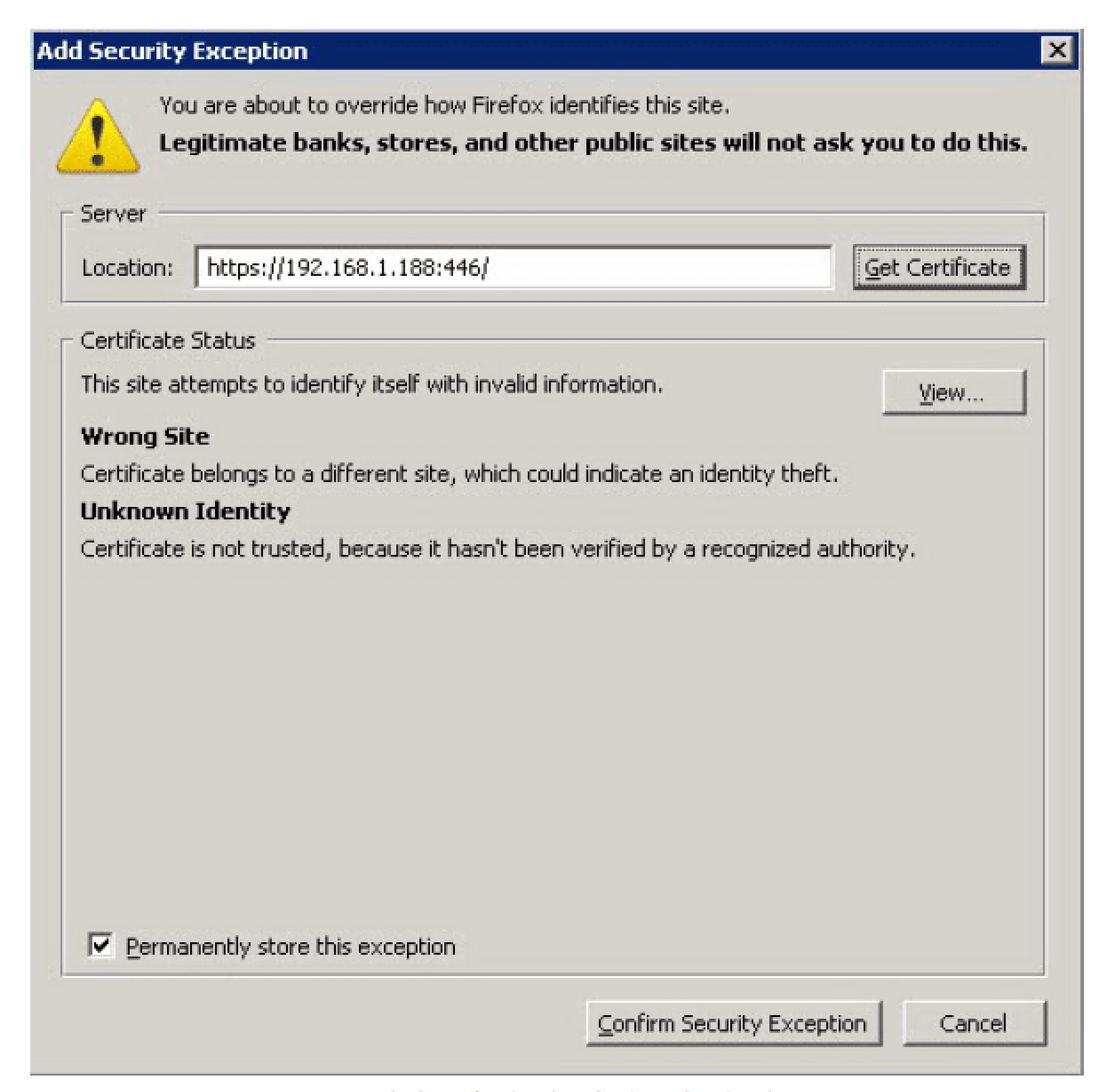


▲ 必须配置例外

3. 弹出如下图所示的画面时，单击 Get Certificate 按钮，并且在弹出如下图所示的画面时，单击 Confirm Security Exception 按钮即可。



▲ 单击 Get Certificate 按钮



▲ 选择确定安全例外的选项

4. 如果你使用 IE 较简单，只要在两个方块中输入 Yes 即可进入。



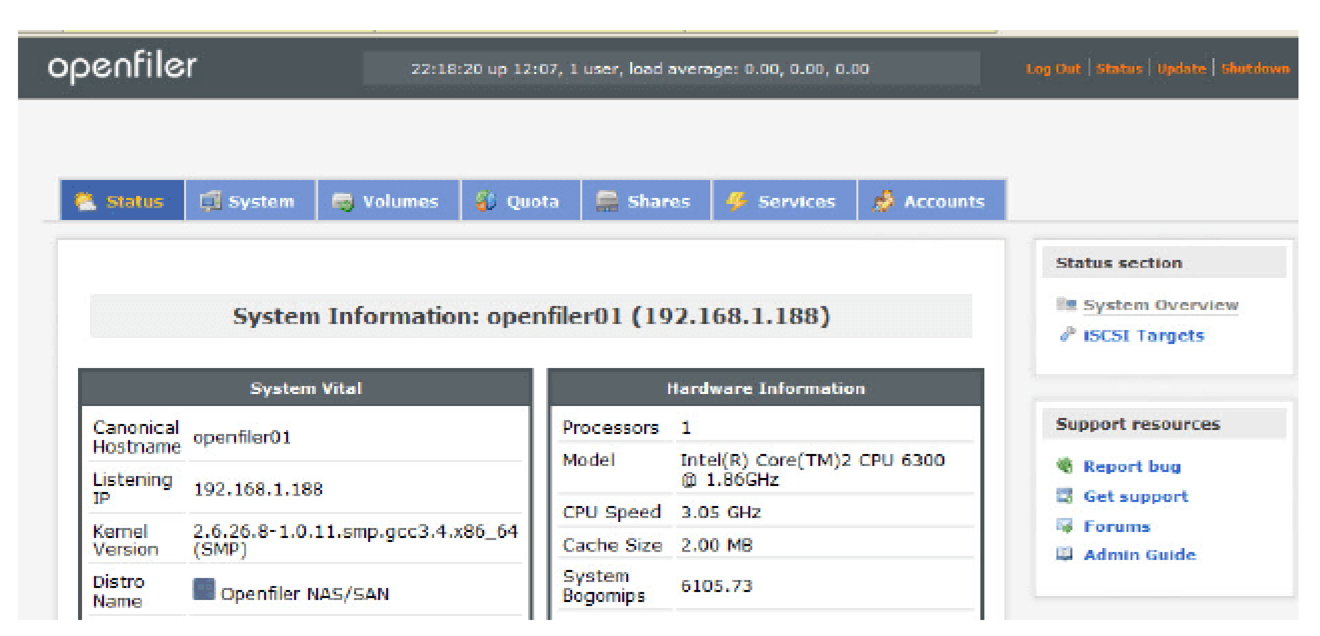
▲ IE 的配置较简单，直接确认后就可以进入

5. 此时会出现 OpenFiler 的登录画面，可以直接输入你的 root 账号和密码，但只能管理本机配置而已，如果要管理存储设备，则要输入 openfiler，密码是 password，即可进入。



▲ 输入账号及密码

6. 进入后，可以看到目前的所有系统状态，如网络接口等，你可以在 Status 和 System 选项卡之下查看和主机有关的功能。



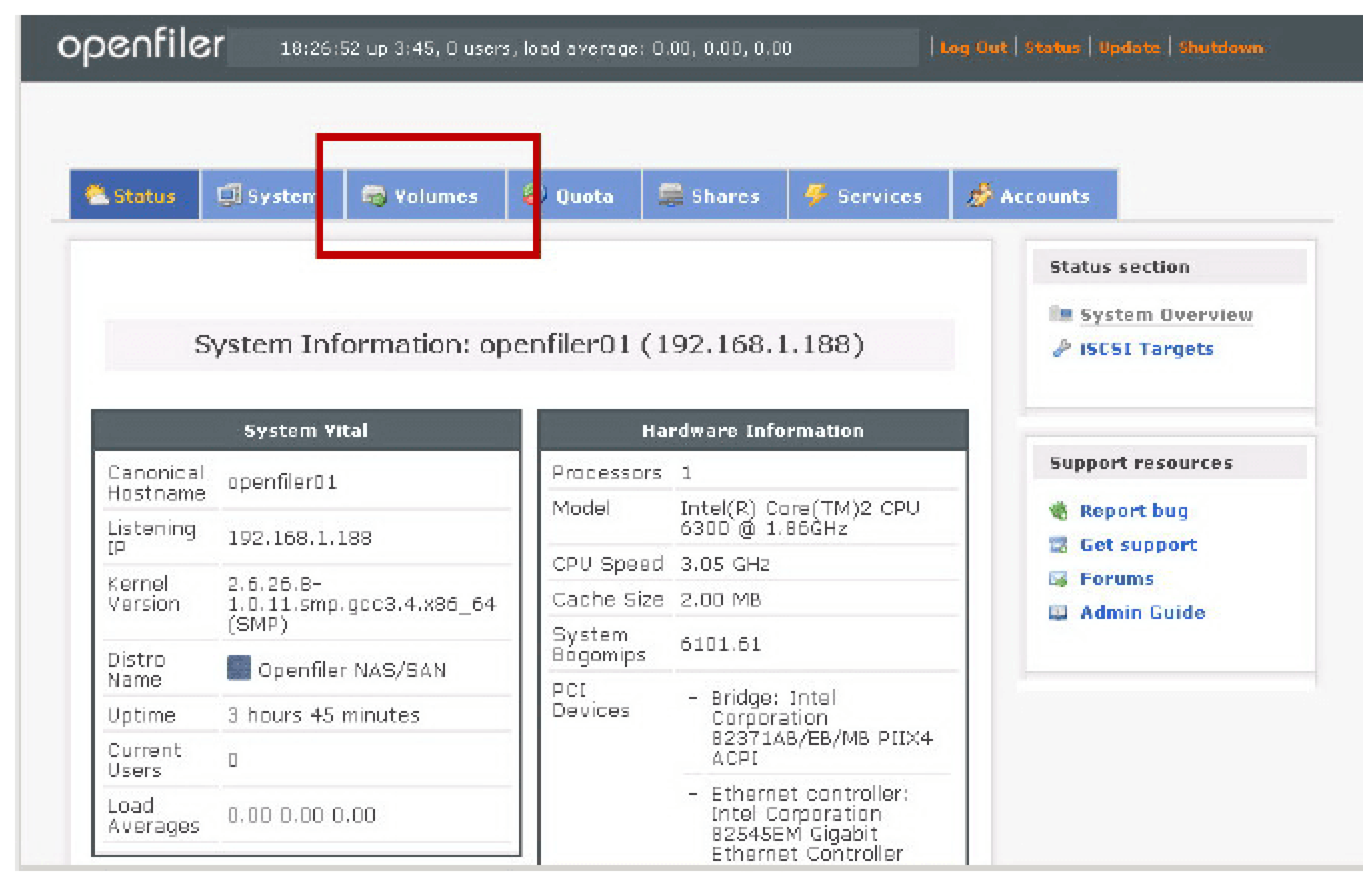
▲ 进入 OpenFiler 的界面

2. 在 OpenFiler 下创建 VG

当我们在硬件层面创建好了两个 RAID 1 的硬盘，并且被 OpenFiler 认出之后，接下来就是要创建 VG。创建 VG 的步骤很简单，我们先做个摘要。

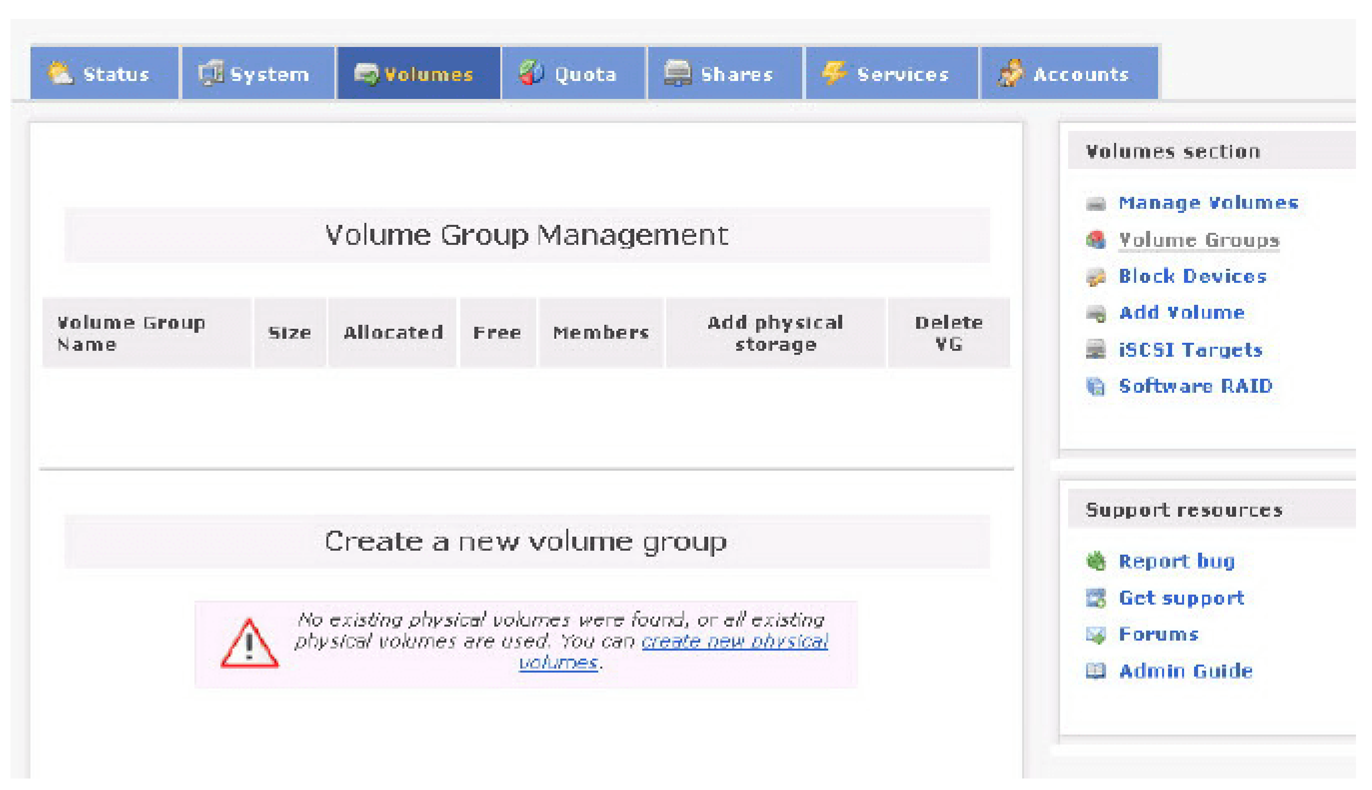
▶ 在 OpenFiler 下创建 PV

1. 首先我们要先找出在 OpenFiler 下的实体硬盘，并且将这两个实体硬盘配置成使用 VG 的格式才行。先进入 OpenFiler，并且选择 Volumes 下的 Manage Volumes。



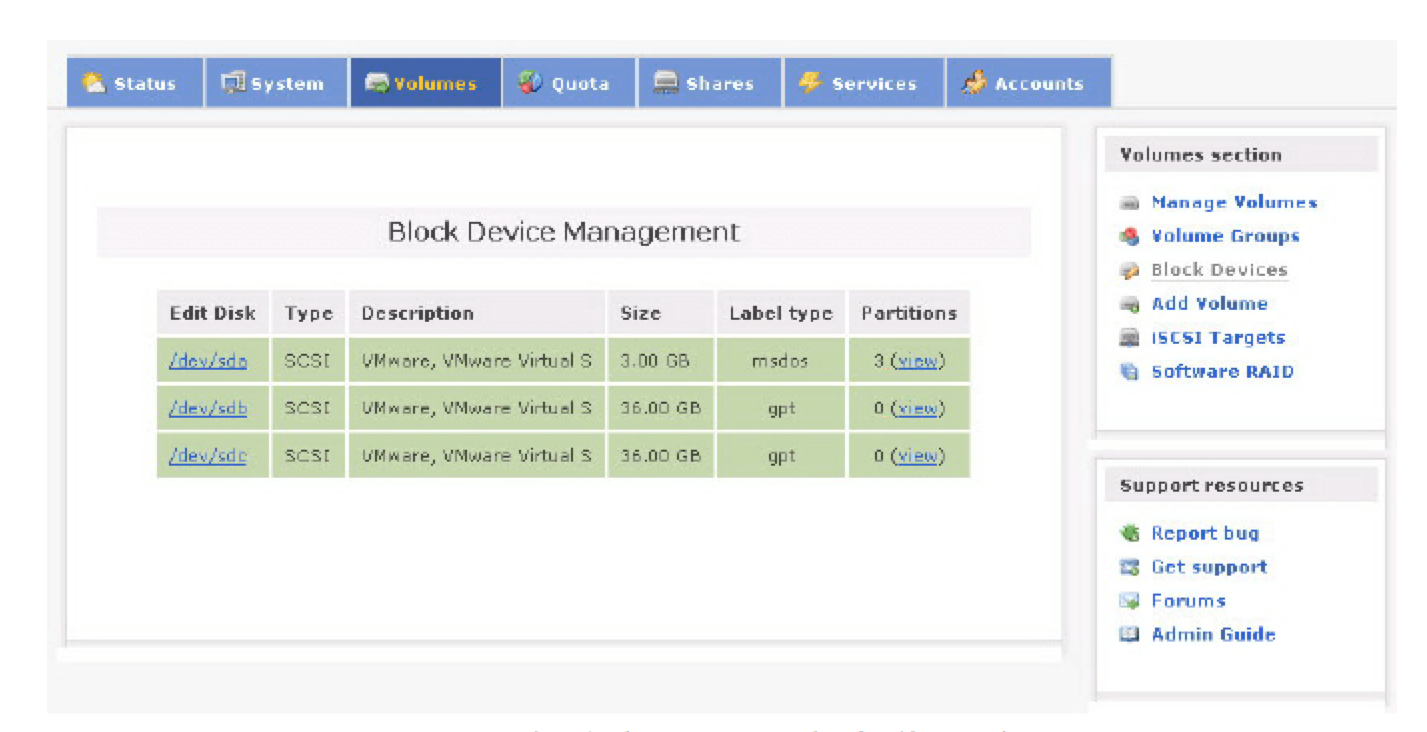
▲ 选择选项

2．在这个画面下，我们可以看到目前还没有任何 VG 存在。此时选择 Block Device 可以检视实体硬盘。



▲ 选择实体硬盘

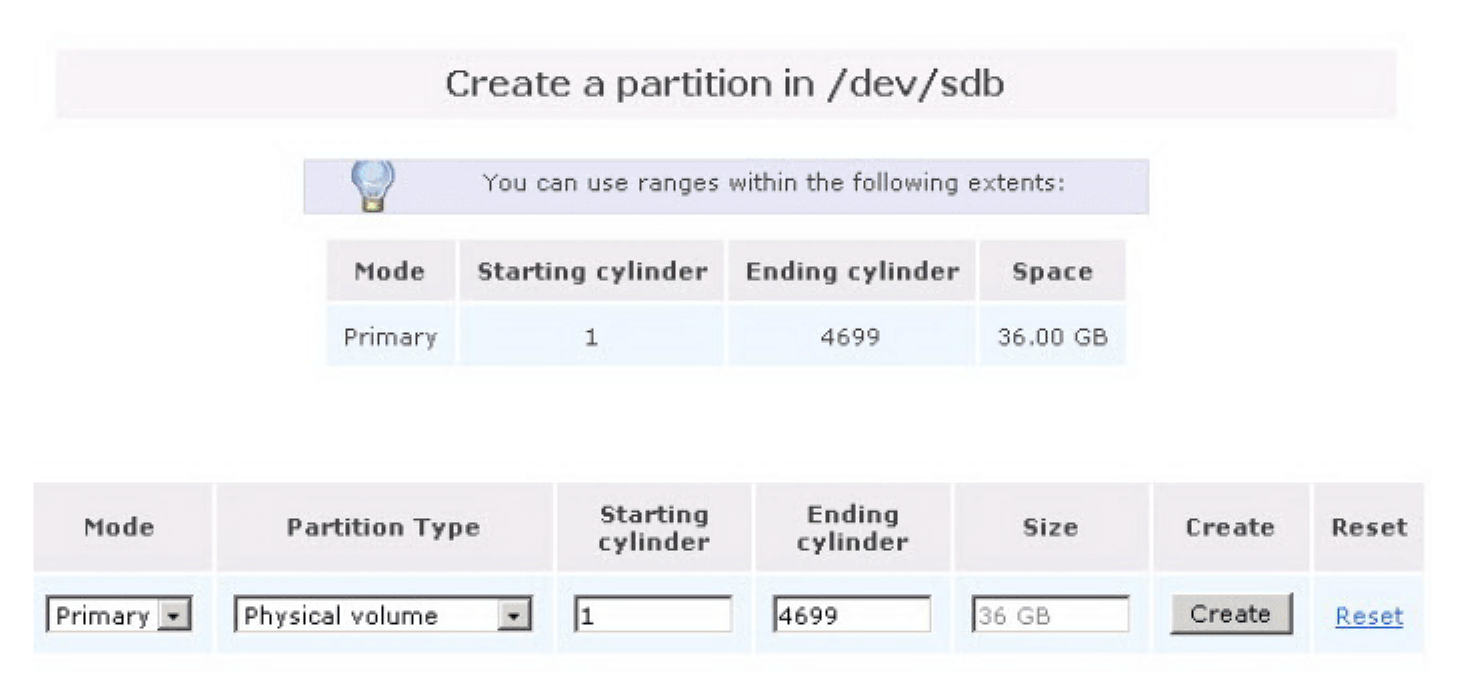
3．由于在这个系统下有三个实体硬盘，第一个已经用于安装系统（/dev/sda），因此我们将第二和第三个硬盘（/dev/sdb、/dev/sdc）组成新的 VG。



Edit Disk	Type	Description	Size	Label type	Partitions
/dev/sda	SCSI	VMware, VMware Virtual S	3.00 GB	msdos	3 (view)
/dev/sdb	SCSI	VMware, VMware Virtual S	36.00 GB	gpt	0 (view)
/dev/sdc	SCSI	VMware, VMware Virtual S	36.00 GB	gpt	0 (view)

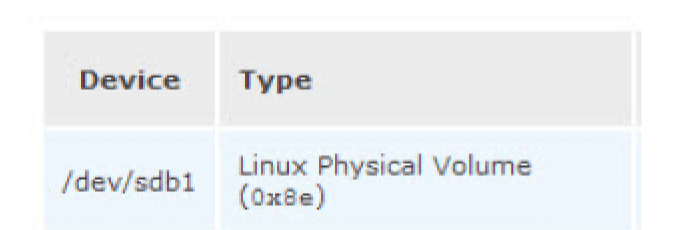
▲ 此时会显示三个实体硬盘

4．首先选择/dev/sdb，此时会进入另一个页面，在这个页面最下方有一个 Create a partition in /dev/sdb。



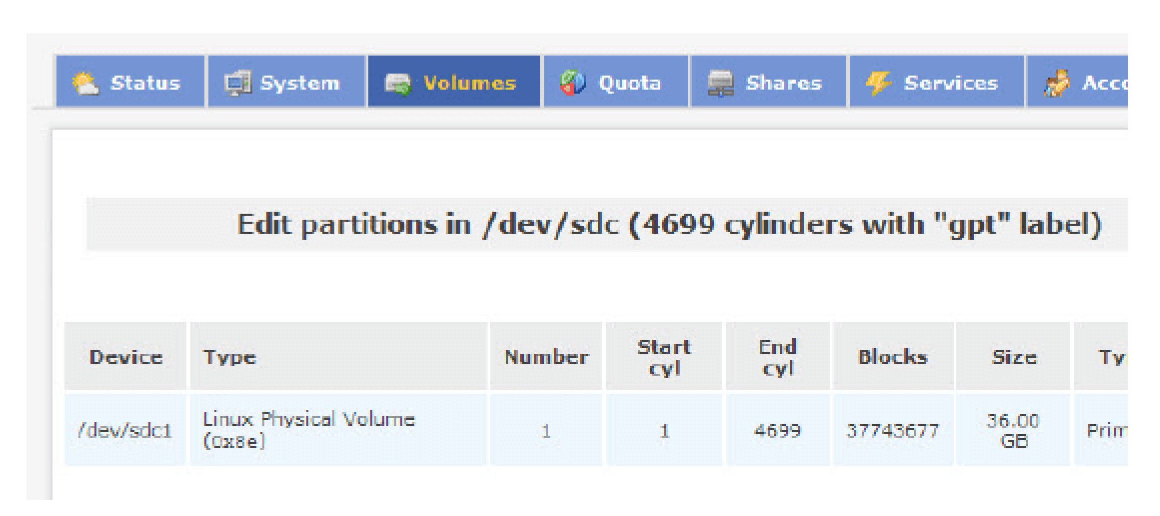
▲ 选择/dev/sdb，这是第一个 SCSI 硬盘

5．在 Mode 的下拉列表框中选择 Primary，Partition Type 中选择 Physical polume。由于我们要使用整个扇区，因此直接单击 Create 按钮即可。



▲ 我们要先将这个硬盘配置成 PV 的格式

6．之后可以看到这个分区已经创建成 PV 的成员之一。重复上面的步骤可以将/dev/sdc 也创建成另一个 PV。



▲ 将第二个也创建好

3. 将 PV 组合成 VG

当你将 PV 创建好之后，就可以利用这些 PV 创建一个 VG。我们就来看看在 OpenFiler 下的创建方法。

▶ 使用 PV 组成 VG

1．首先选择 Manage Volumes。



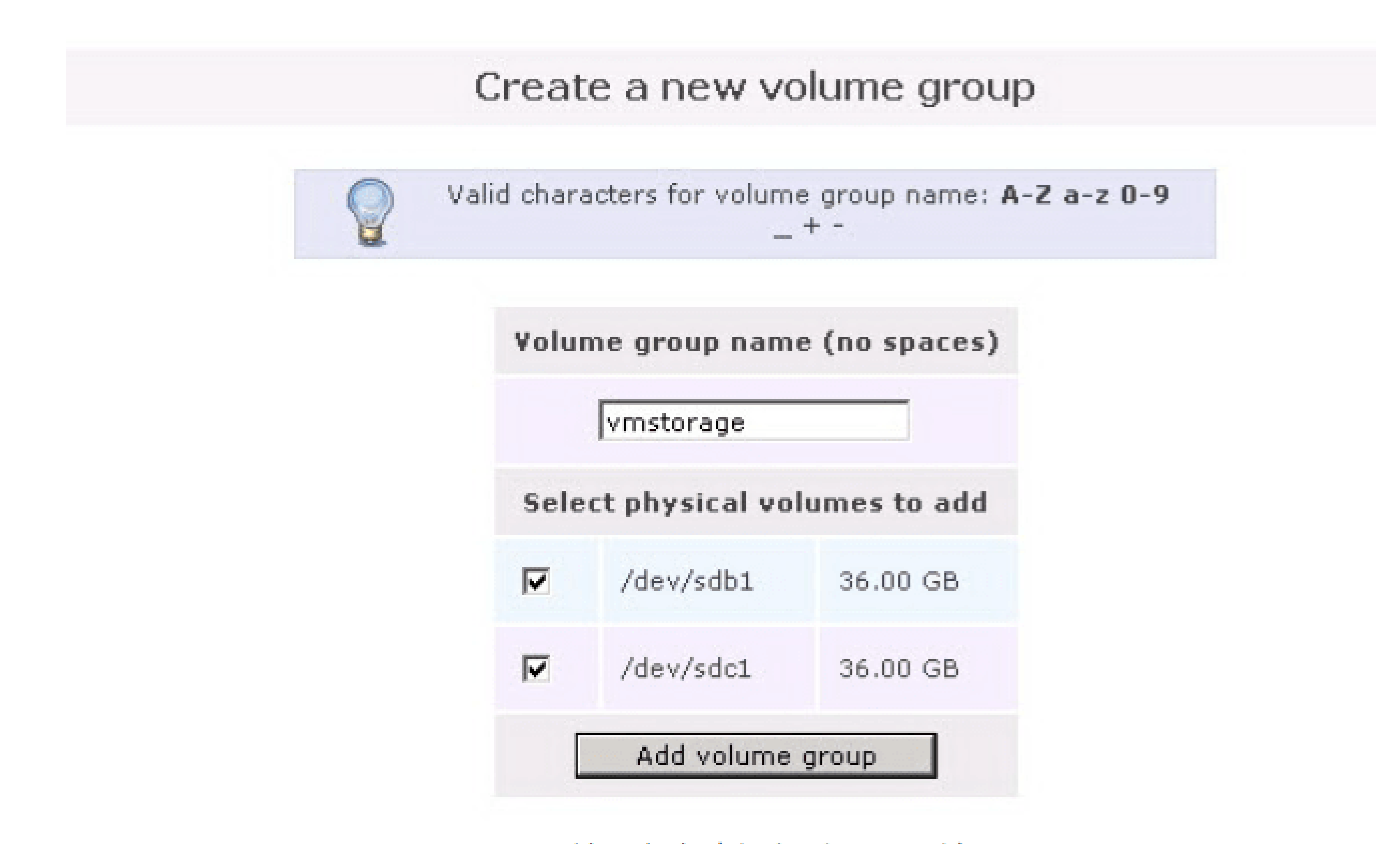
▲ 选择选项

2．输入名称，如 vmstorage。



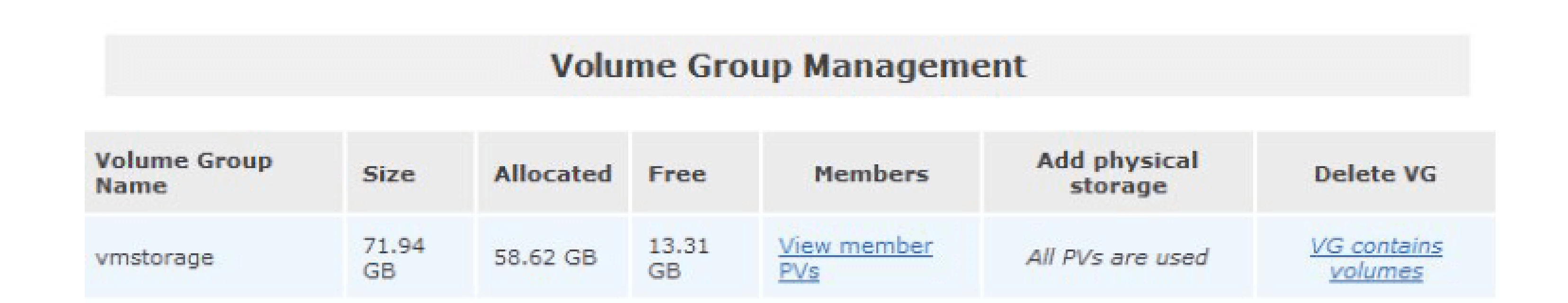
▲ 创建 Volume Group

3．将两个 PV 选中，并且单击 Add volume group 按钮。



▲ 将刚才创建的 PV 放入

4．此时我们可以看到在 Volume Group Management 中列出我们创建好的 VG 了。



Volume Group Management

Volume Group Name	Size	Allocated	Free	Members	Add physical storage	Delete VG
vmstorage	71.94 GB	58.62 GB	13.31 GB	View member PVs	All PVs are used	VG contains volumes

▲ 这个 VG 就完成了

12.2.3 创建 iSCSI 的分区和连接

VG 组合完毕之后，我们可以为整个大的 VG 划分逻辑分割区，在一般的 SAN 设备上，这些逻辑分割区都称为 LUN，而在 OpenFiler 上则称为 LV。本章的重点是划分出具备 iSCSI 连接能力的 LUN，因此步骤是先做出 LUN，之后再开启 iSCSI Target 功能，并且指定 iSCSI 的参数，步骤为：

▶ 开启 iSCSI 的步骤

（1）开启 iSCSI 功能。

（2）创建 iSCSI 的 LUN。

（3）配置访问 iSCSI 的 IP 网络。

（4）配置 iSCSI 的参数。

（5）使用 iSCSI 的客户端测试是否成功。

1. 开启 iSCSI Target Server 功能

在 OpenFiler 中，默认是将 iSCSI Target Server 的功能关闭，因此我们要使用操作系统作为 iSCSI SAN，必须先将其打开，只要到 System 选项卡下，将原来 Disabled 的 iSCSI target server 的 Enable 按钮单击即可。当然你也可以在这里开启其他的功能，包括 FTP、SMB、NFS 等。

Manage Services

Service Name	Status	Modification
SMB / CIFS server	Disabled	Enable
NFSv3 server	Disabled	Enable
HTTP / WebDAV server	Disabled	Enable
FTP server	Disabled	Enable
iSCSI target server	Disabled	Enable
Rsync server	Disabled	Enable
UPS server	Disabled	Enable
LDAP server	Disabled	Enable
ACPI daemon	Enabled	Disable
iSCSI initiator	Enabled	Disable

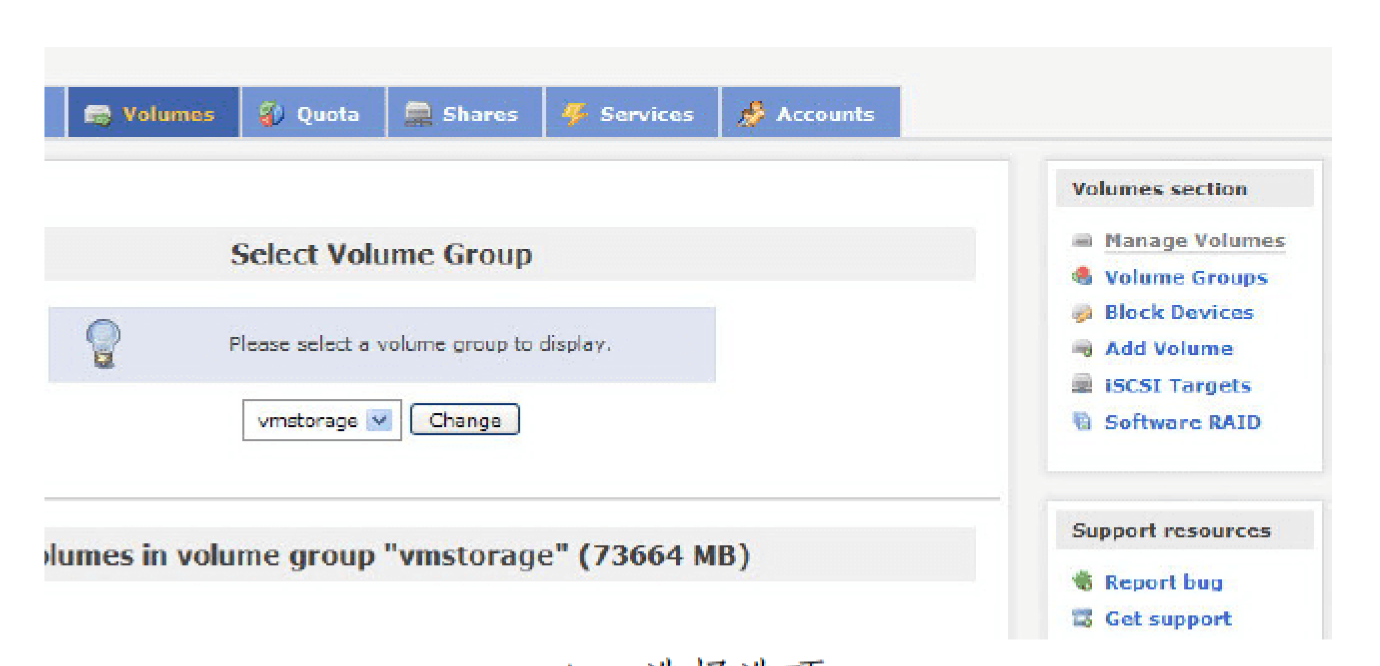
▲ 在这里可以将需要的功能开启

2. 创建 iSCSI 逻辑分区 LUN

在独立存储设备中，LUN（Logical Unit Number）是最重要的基本单位。LUN 可以被 SAN 中的任何主机访问，不管是透过 HBA 或是 iSCSI。就算是软件激活的 iSCSI，也可以在不同的操作系统之下，在操作系统启动之后利用软件的 iSCSI initiator 访问 LUN。在 OpenFiler 之下，LUN 被称为 Logical Volume（LV），因此在 OpenFiler 下创建 LUN 就是创建 LV。

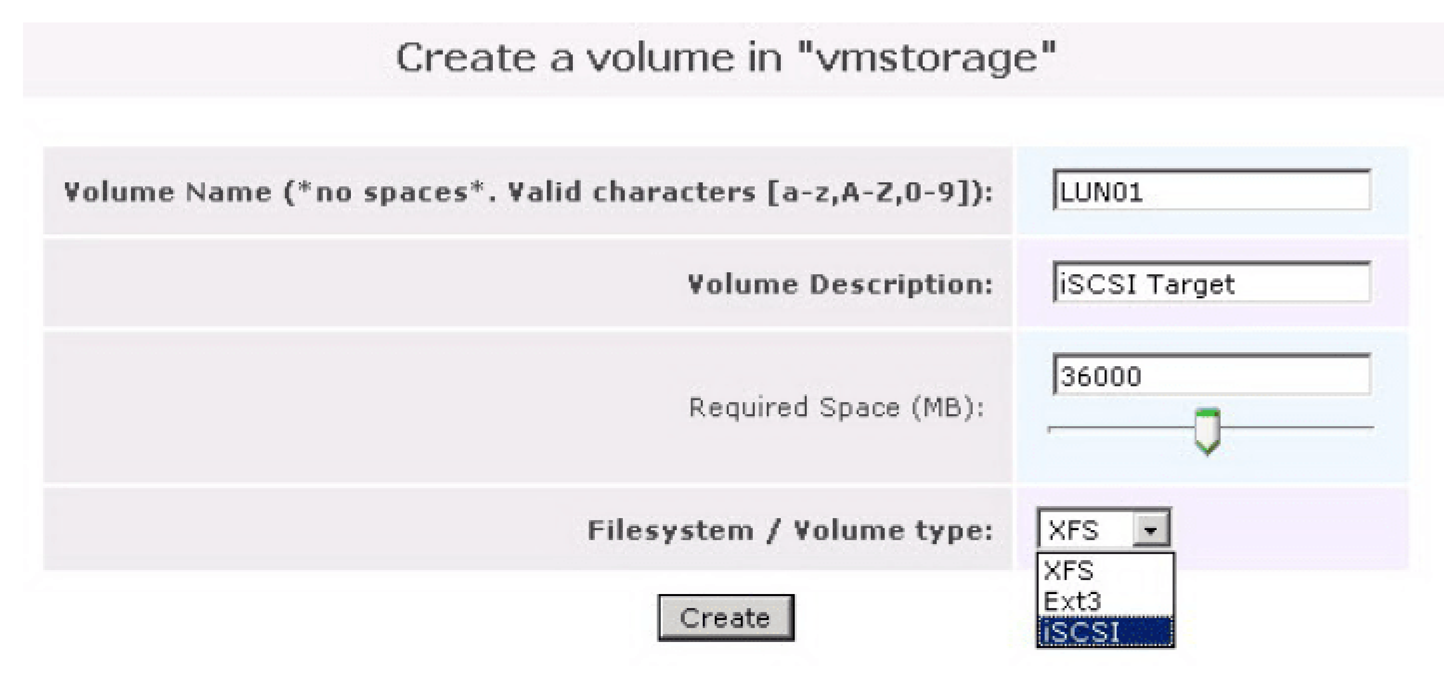
▶ 在 OpenFiler 下创建 iSCSI 的 LUN

1. 进入 OpenFiler 的 Web 接口，选择 Volumes 中的 Manage Volumes。
2. 选择 Volume Groups，可以看到目前已经创建的 vmstorage 这个 VG，目前上面还没有任何 LUN。



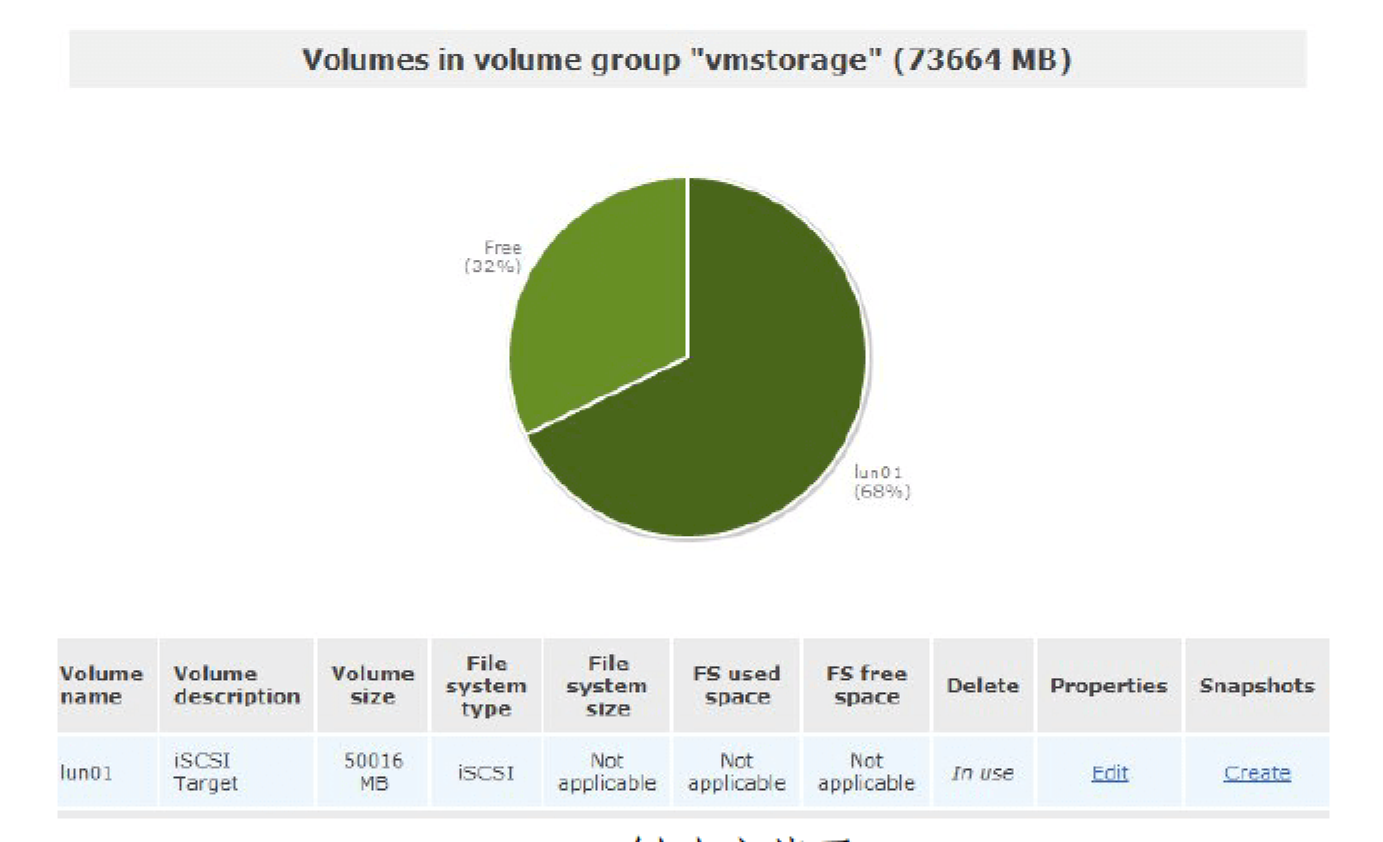
▲ 选择选项

3. 单击 Add Volume 按钮，就会进入加入新 LV 的画面。最下面有一个创建 LV 的地方，我们输入 LV 的名称（如 LUN01），输入描述、大小，并且在 Filesystem/volume type 下拉列表框中选择 iSCSI。之后单击 Create 按钮。



▲ 选择添加卷，并且输入大小和说明

4．至此 iSCSI 的 LUN 已经创建完毕了。



Volume name	Volume description	Volume size	File system type	File system size	FS used space	FS free space	Delete	Properties	Snapshots
lun01	iSCSI Target	50016 MB	iSCSI	Not applicable	Not applicable	Not applicable	In use	Edit	Create

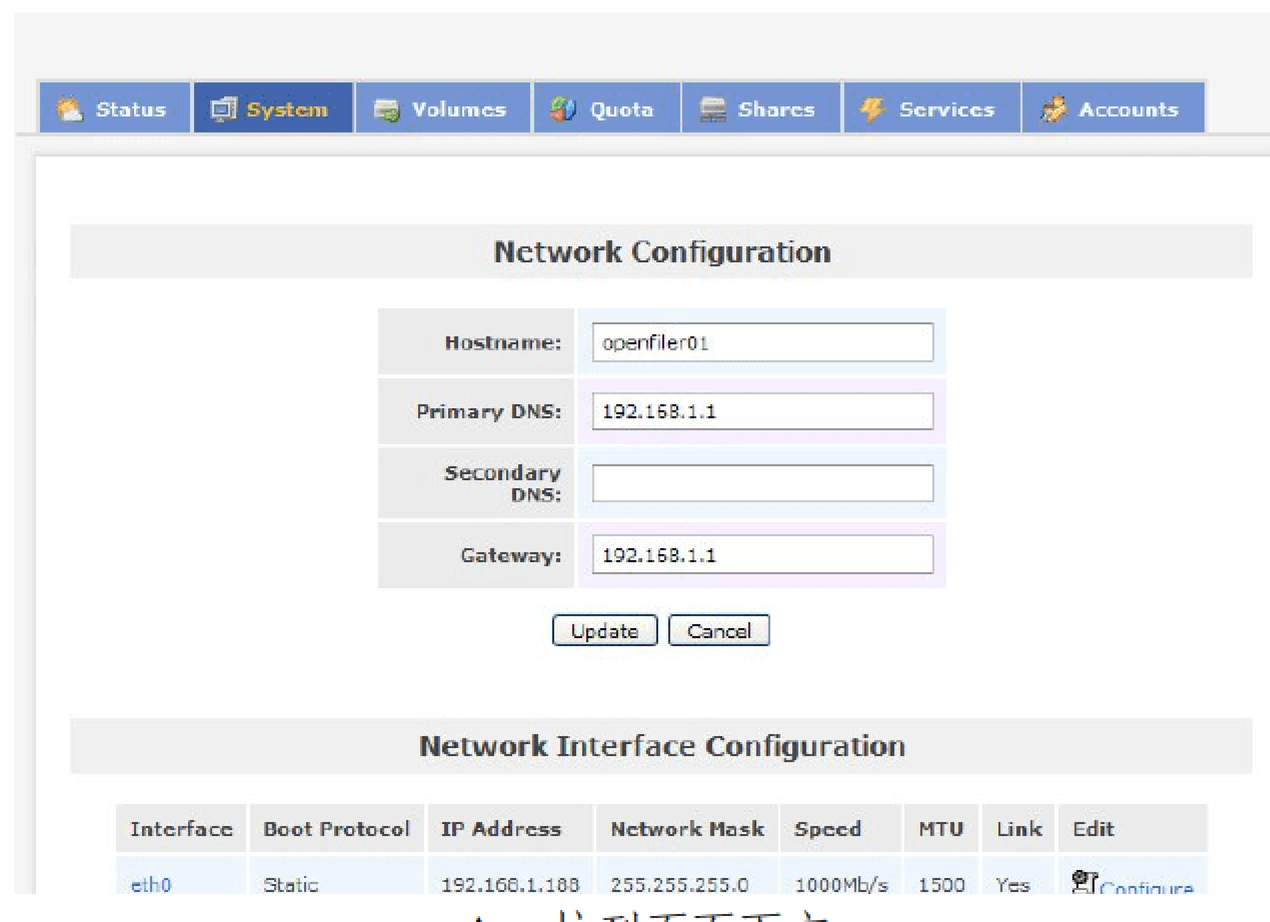
▲ 创建完毕了

3. 开启 IP 网络

由于 iSCSI 是走 IP 网络，因此我们要允许网络中的计算机可以透过 IP 来访问。下面就是 OpenFiler 中 IP 网络和同一网段中其他主机的连接方法。

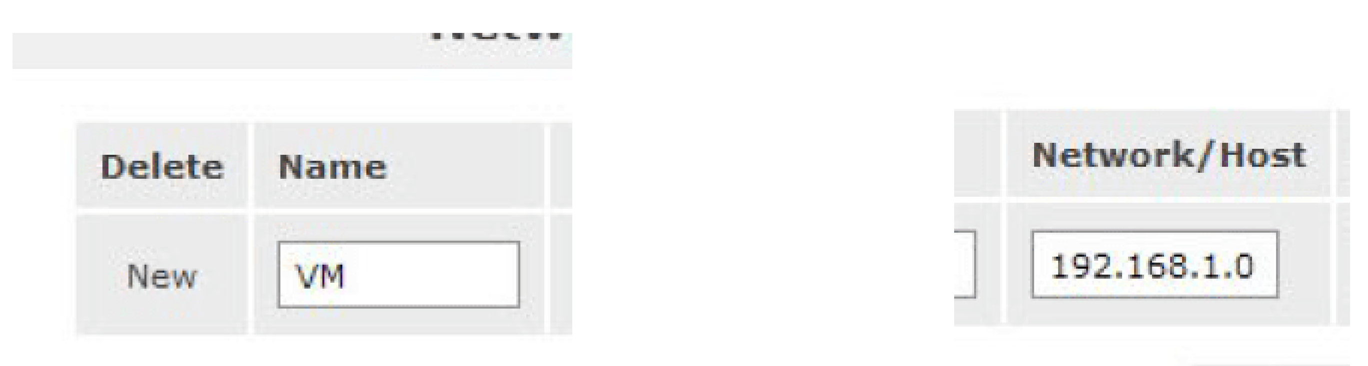
▶ 配置 OpenFiler 中的 IP 网段

1．进入 OpenFiler 中的 System，并且直接拉到页面的下方。



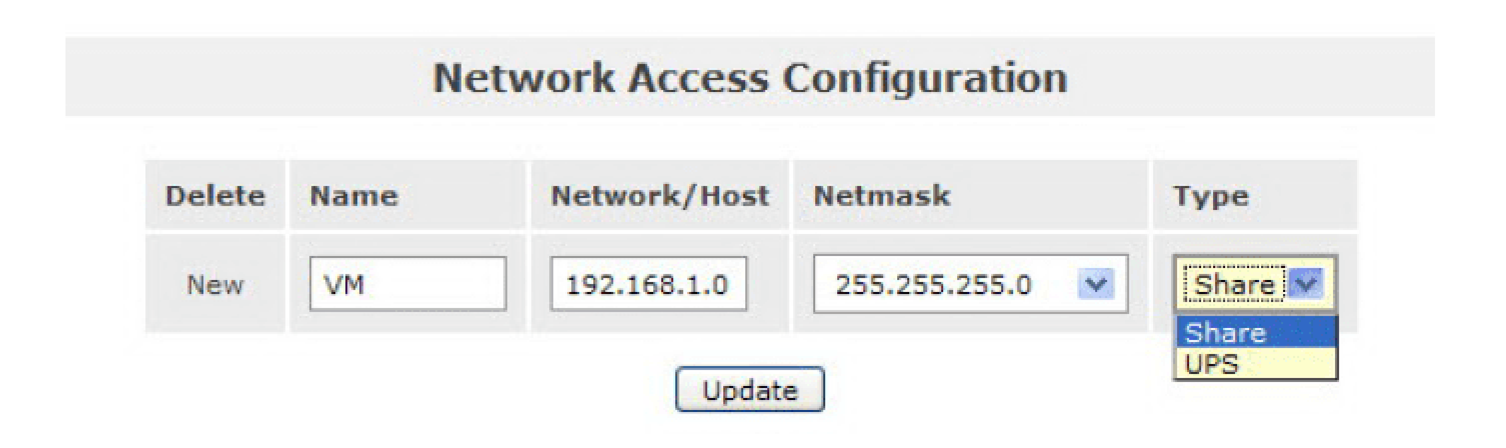
▲ 拉到页面下方

2. 在 Network Access Configuration 的地方输入这个网络访问的名称，如 VM。
3. 输入主机的 IP 段。注意不可以输入单一主机的 IP，这样会都无法访问。我们在这边输入 192.168.1.0，表示从 192.168.1.1 一直到 192.168.1.254 都能访问。



▲ 给这个网段一个名称，如 VM　　▲ 输入网段 IP 而非主机 IP

4. 在 Netmask 中选择 255.255.255.0，并且在 Type 下拉列表框中选择 Share，之后即可以单击 Update 按钮。



▲ 选择完之后就更新

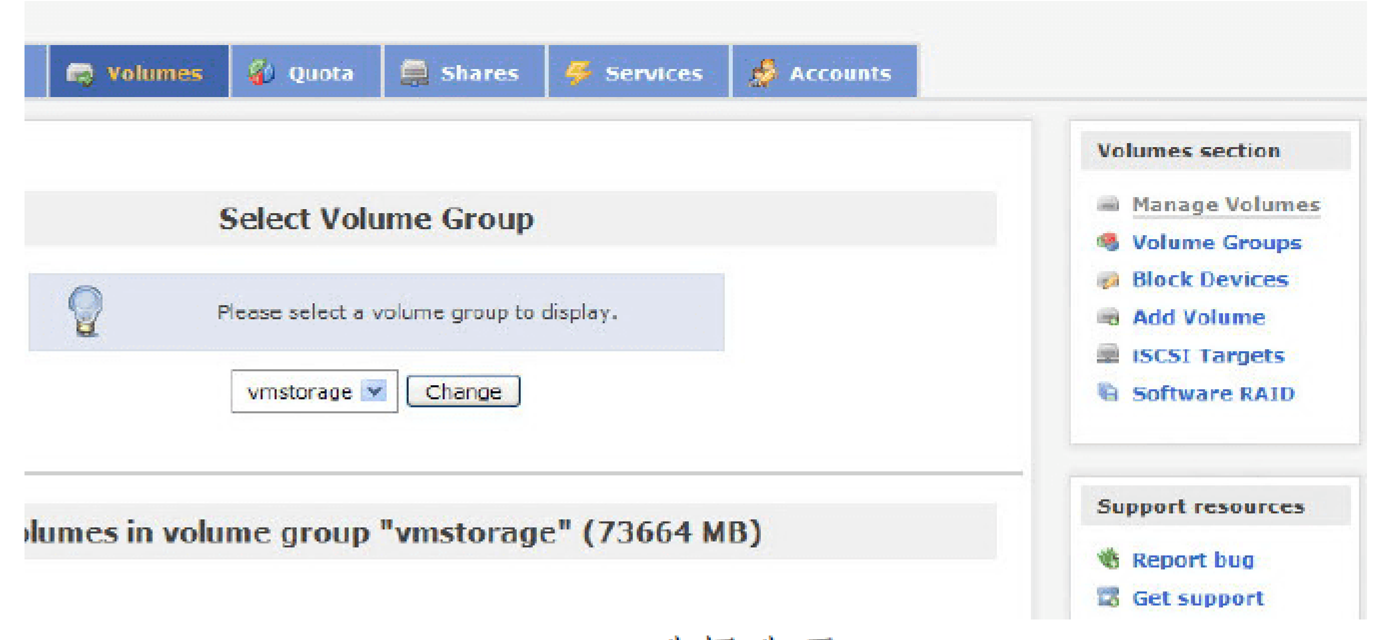
5. 至此就可以在这个 OpenFiler 中看到被授权的网段了。

4. 配置 iSCSI 参数

配置 iSCSI 参数主要是让客户端能利用这个参数访问到 LUN。在 OpenFiler 中，配置参数的动作很简单，主要就是先添加 iSCSI 的设备号码，并且将 LUN 映像到这个号码上，最后再经过安全配置，就可以连接了。

▶ 配置 iSCSI 连接参数

1. 还是进入 OpenFiler 的 Volumes 中，并且选择 iSCSI Targets。



▲ 选择选项

2. 此时会出现 Add new iSCSI Target。你会看到有一个 Target IQN 的字段，这个字段称为 iSCSI 合格证（iSCSI Qualified Number），是每一个 iSCSI 唯一的编号，也是在网络上辨认 iSCSI 设备的唯一编号。这个号码由系统产生，我们可以不需要更动。在此单击 Add 按钮。



▲ 这里要加入 iSCSI 的合格编号

3．在新增之后，我们可以在方块下方看到完整的 IQN 参数，这个参数暂时不用更动，使用默认值即可。



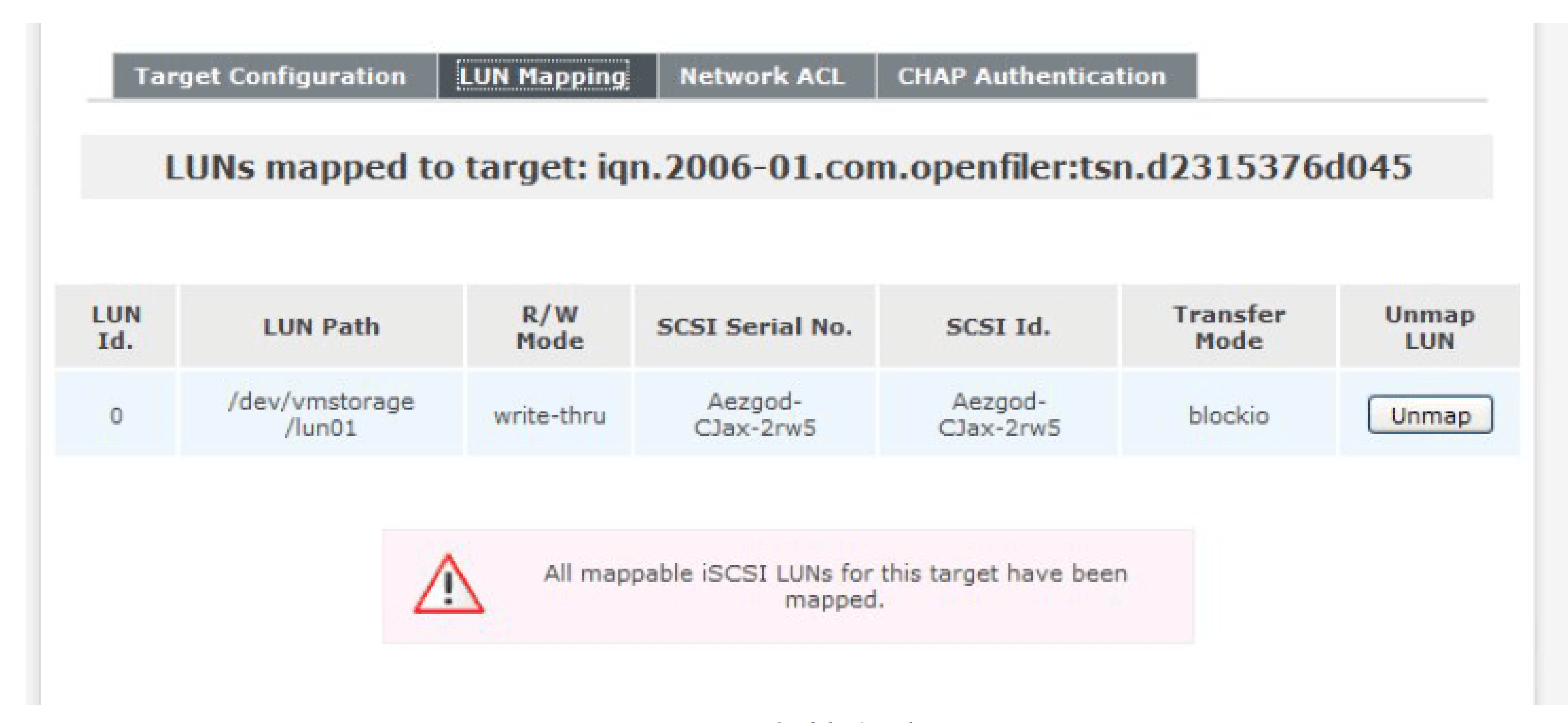
▲ 这边有 iSCSI 的使用选项

4．在 IQN 创建之后，接下来我们要做的就是将 LUN 映射到这个 IQN 上，动作也很简单，因为 OpenFiler 都帮我们做好了。只要进入 LUN Mapping 的选项卡，选择你刚才创建的 LUN（如图中的 lun01），并且单击 Map 按钮即可。



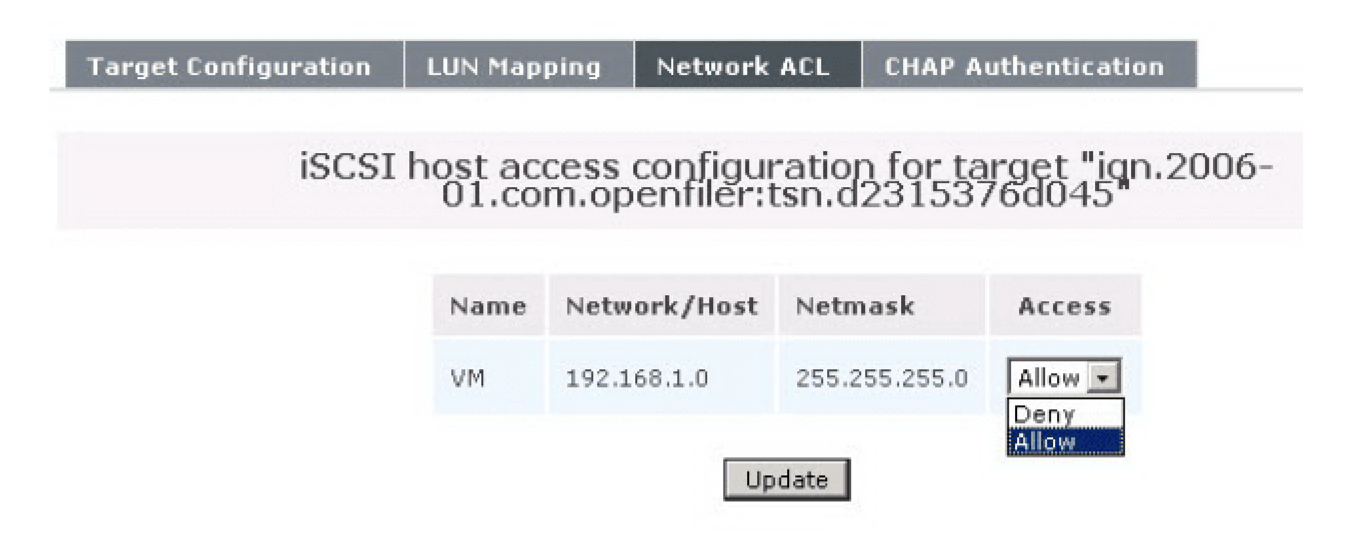
▲ 选择映射 IQN 到刚才的 LUN 上

5．映射完成后，会出现如下图所示的的画面，随时可以再 Unmap。



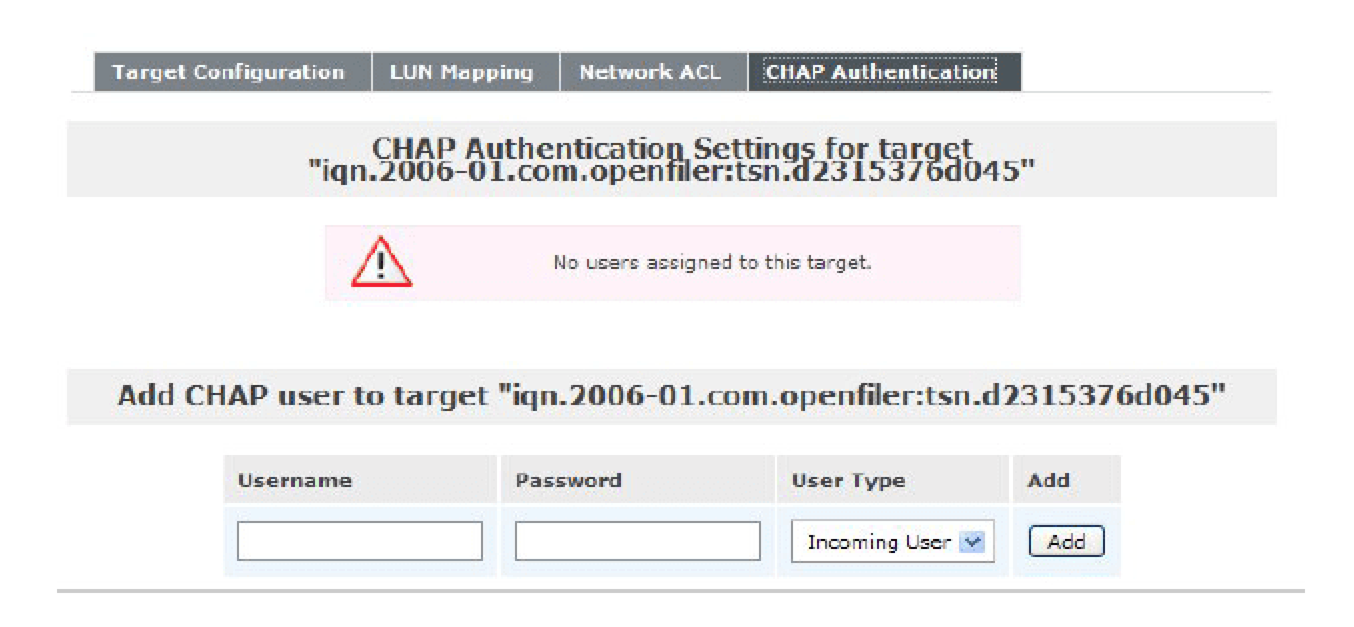
▲ 映射完毕

6. 此时选择 Network ACL 选项卡，将 Access 改成 Allow，如下图所示，并且单击 Update 按钮。



▲ 要让网络上的用户可以读取，要改成 Allow

7. 如果你希望系统更为安全，可以在 CHAP 之中加入账号及密码，我们在此先不配置这个项目，使用网段的限制即可。



▲ 如果需要更安全的配置，可以使用 CHAP

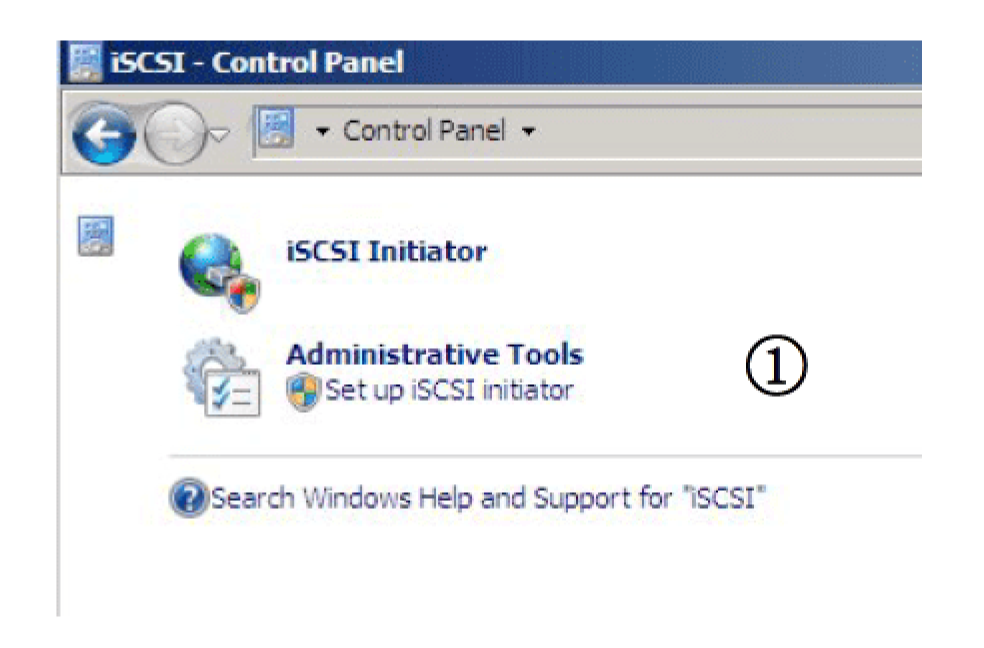
8. 到此，你的 iSCSI 已经完全可以使用了，只要你的操作系统安装了 iSCSI initiator，不管是软件还是硬件的。

5. 使用 Windows 系统测试 iSCSI 的连接

在安装 VMware 或 Hyper-V 之前，我们可以使用 XP、Vista 或是 Windows Server 2008 等系统来测试这个 iSCSI 的连接是否成功，尤其是在 Vista 或是 Windows Server 2008 下，操作系统本身就附有 iSCSI initiator，而 XP 则是需要安装，我们就来看看 XP 操作系统下的连接方式。

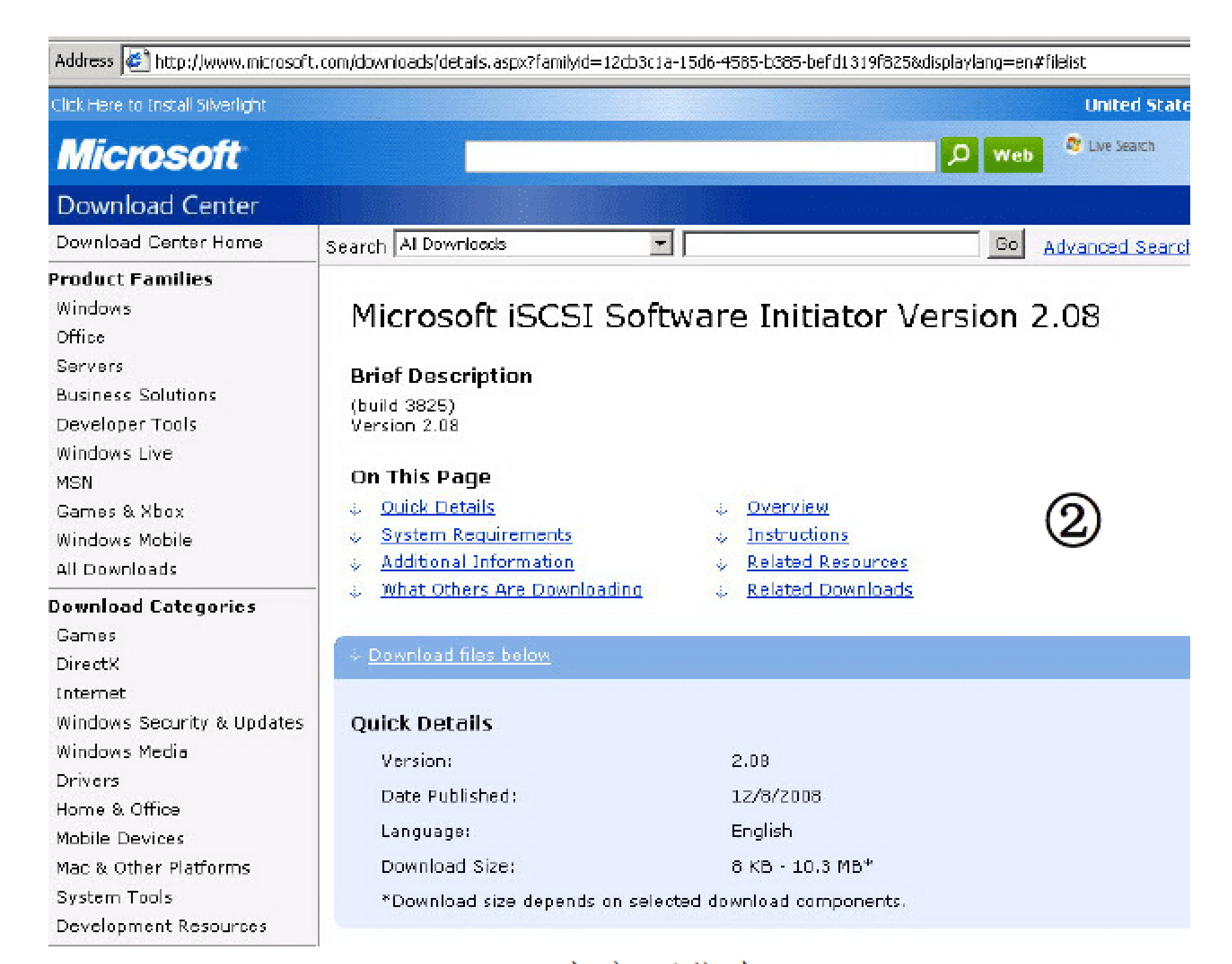
▶ 下载及安装 XP 的 iSCSI Initiator

1. 如果你使用的是 Vista 或是 Windows Server 2008，则在控制台中就已经有 iSCSI 的连接器。



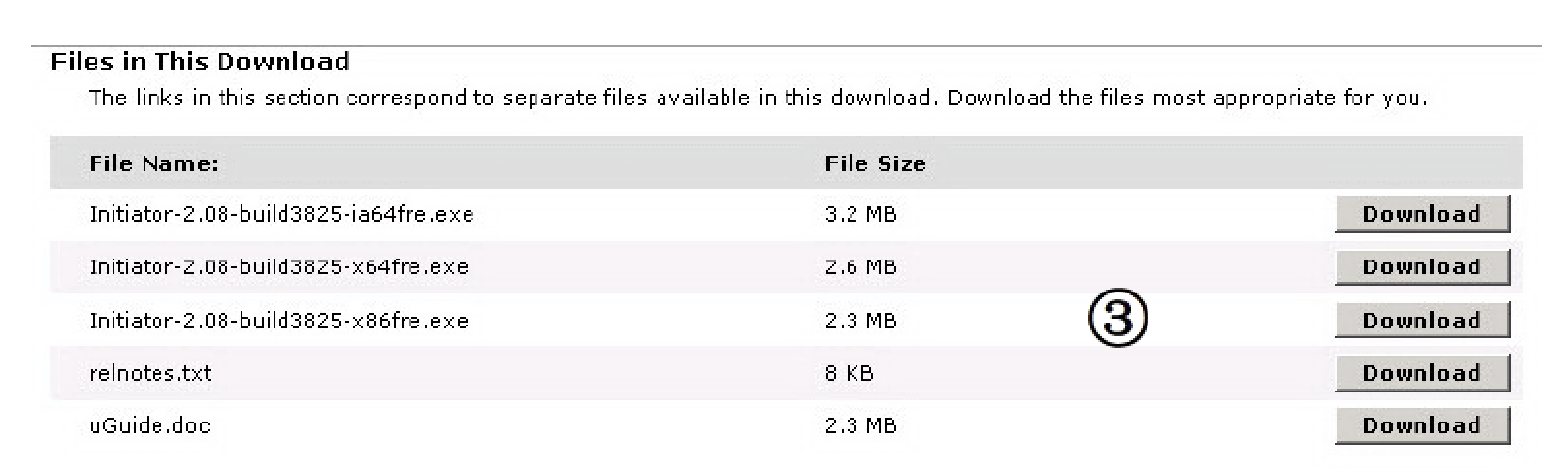
▲ 在 Vista 或是 Windows Server 2008 已经有 iSCSI 的连接器了

2．如果使用 XP，则必须去微软下载 iSCSI initiator for XP。我们直接到微软的官方网页下载，在 Google 中直接输入 XP iSCSI 即可找到下载点。



▲ 官方下载点

3．下载时选择你的 XP 版本，大部分都是 32 位，下载 X86 版本即可。



Files in This Download

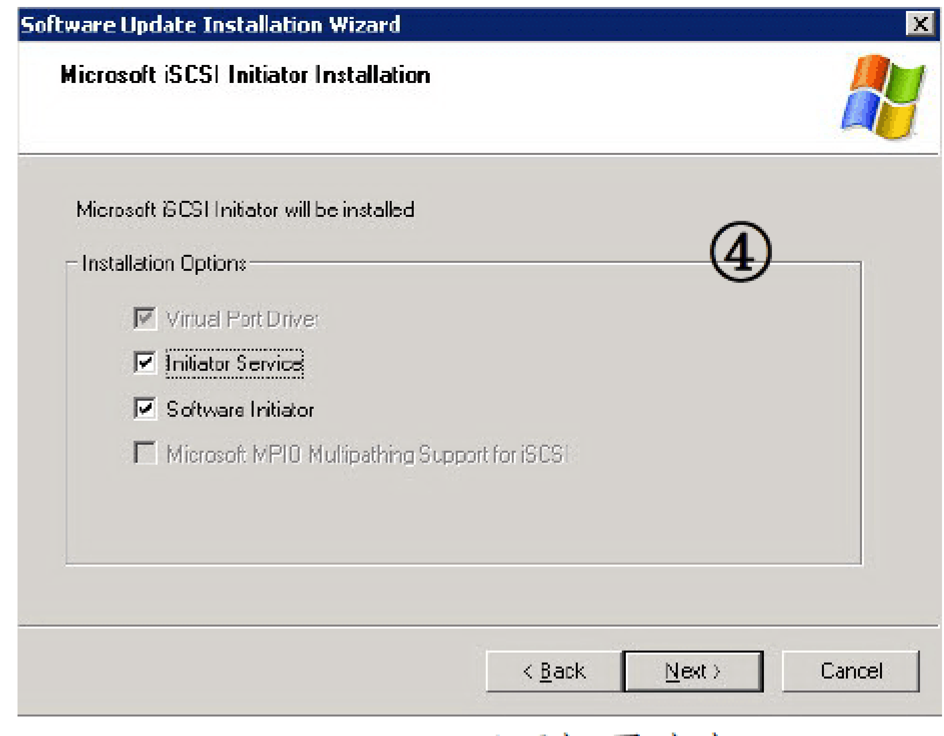
The links in this section correspond to separate files available in this download. Download the files most appropriate for you.

File Name:	File Size	
Initiator-2.08-build3825-ia64fre.exe	3.2 MB	Download
Initiator-2.08-build3825-x64fre.exe	2.6 MB	Download
Initiator-2.08-build3825-x86fre.exe	2.3 MB ③	Download
relnotes.txt	8 KB	Download
uGuide.doc	2.3 MB	Download

▲ 选择你的版本，通常使用 XP 都是 32 位版本的

4．下载回来后直接单击开始安装，第一个画面中，我们两项都要选中。

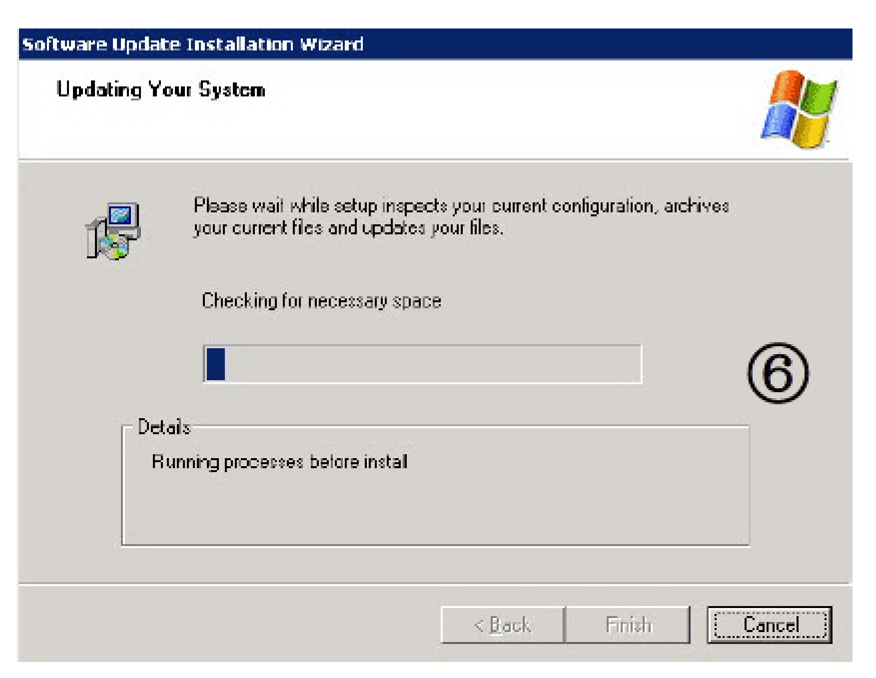
5．同意后单击 Next 按钮。



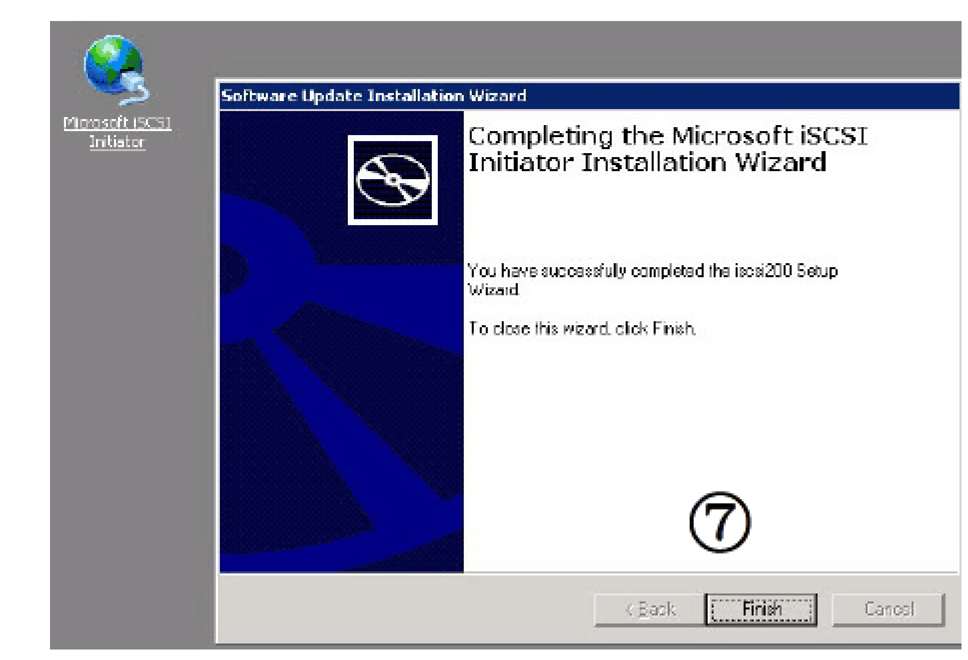
▲ 两项都要选中　　　　▲ 开始安装

6．接下来将开始安装。

7．当安装完毕会弹出如下图所示的画面，在桌面上也会有连接图示。



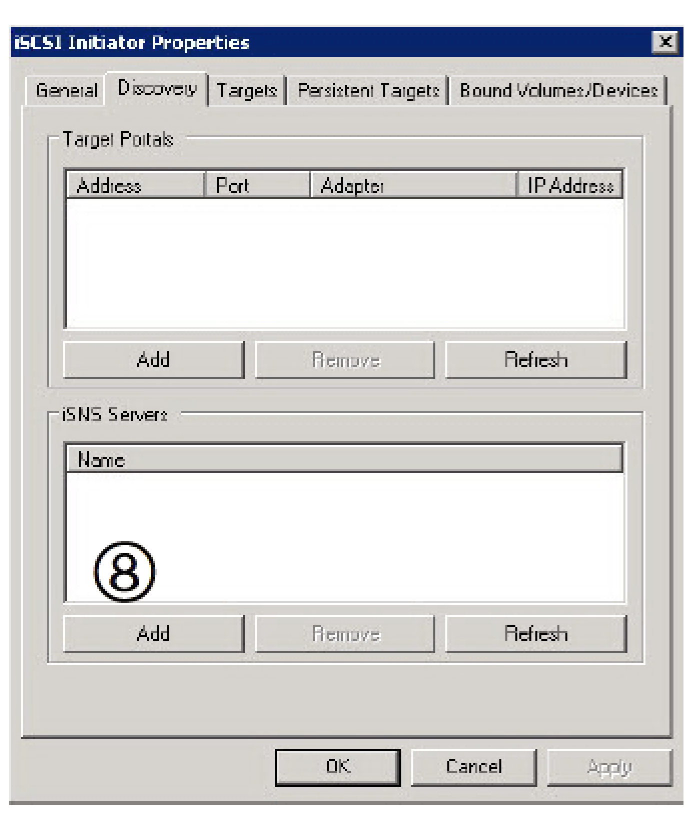
▲ 主要是在 Windows 中加上 iSCSI initiator 的服务



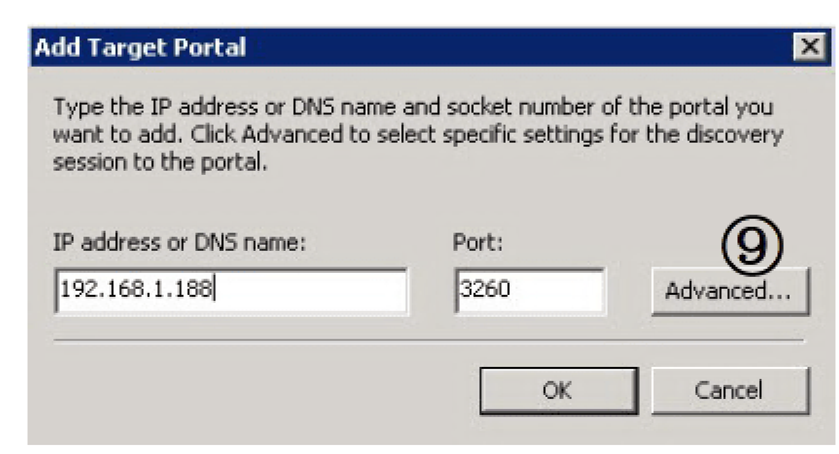
▲ 安装完毕

8．进入该图示，由于没有任何连接，因此这里会是空的。选择 Discovery 选项卡，并且单击 Add 按钮来新增 iSCSI Target。

9．输入安装 OpenFiler 的 IP，直接单击 OK 按钮即可。



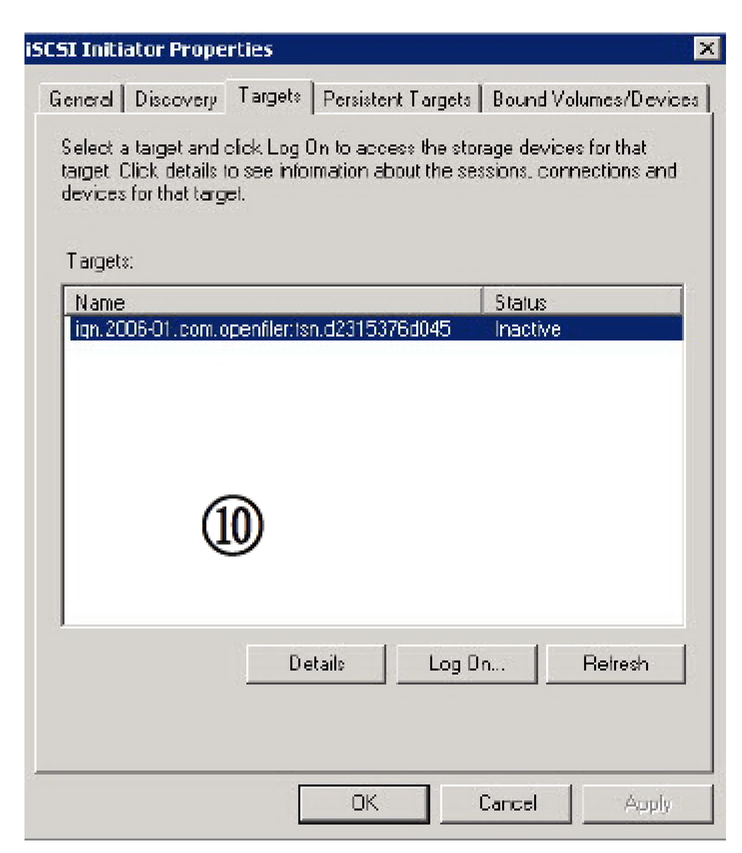
▲ 选择 Discover 来输入 Target Server 的 IP



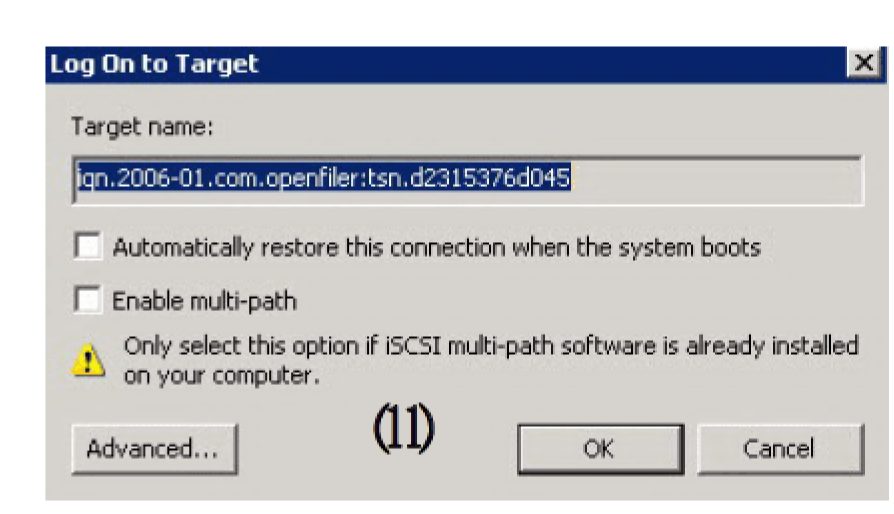
▲ 在这里输入 IP

10．此时在 Targets 选项卡中，就会出现我们对映的 LUN，但这个时候的状态是 Inactive，即无法使用，单击 Log On...按钮登录。

11．当出现如下图所示的画面时，表示我们要将这个 LUN 加入计算机中，单击 OK 按钮即可。



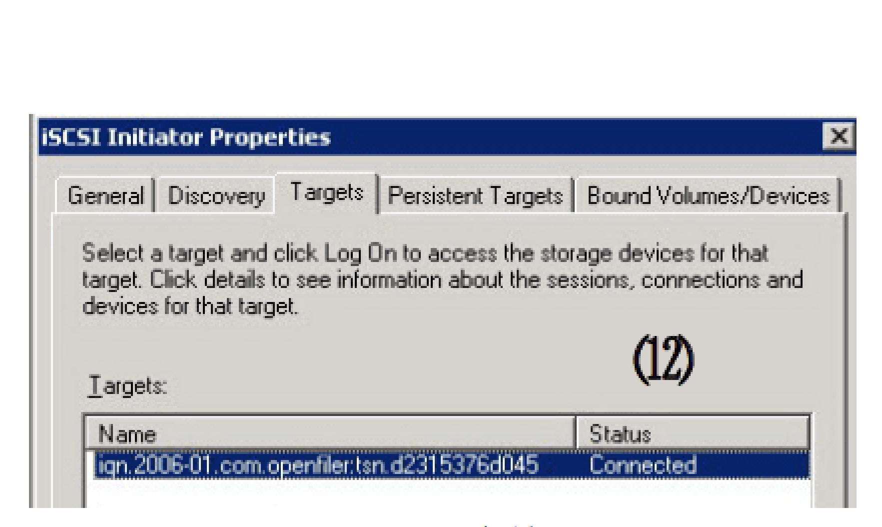
▲ 虽然映射成功，但还没登录无法使用，要单击 Log On 按钮登录



▲ 选择要登录的 LUN

12．当加入之后，可以看到状态已经是连接上了。

13．此时我们回到 OpenFiler 的网页接口上，也可以看到连接的情况。



▲ 已连接了



▲ 连接完毕，可以在 Web 接口上看到连接情况

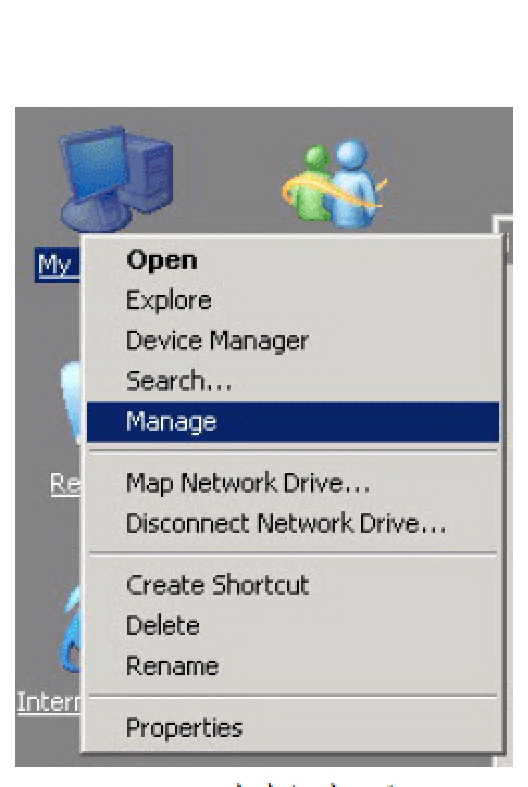
6. iSCSI 在 XP 下的基本分享单位

常常使用共享磁盘的人都知道，不管是 Windows 的或是 Linux 的分享，基本上都是使用共享文件夹。但是在 iSCSI 的使用时，我们竟然可以用到分割区的层次。可以在上面安装不同的文件系统，并且可以针对不同的需要进行分享，如文件层次的共享文件夹、Cluster 层次的 quorum 硬盘等，在本书的稍后会有介绍。

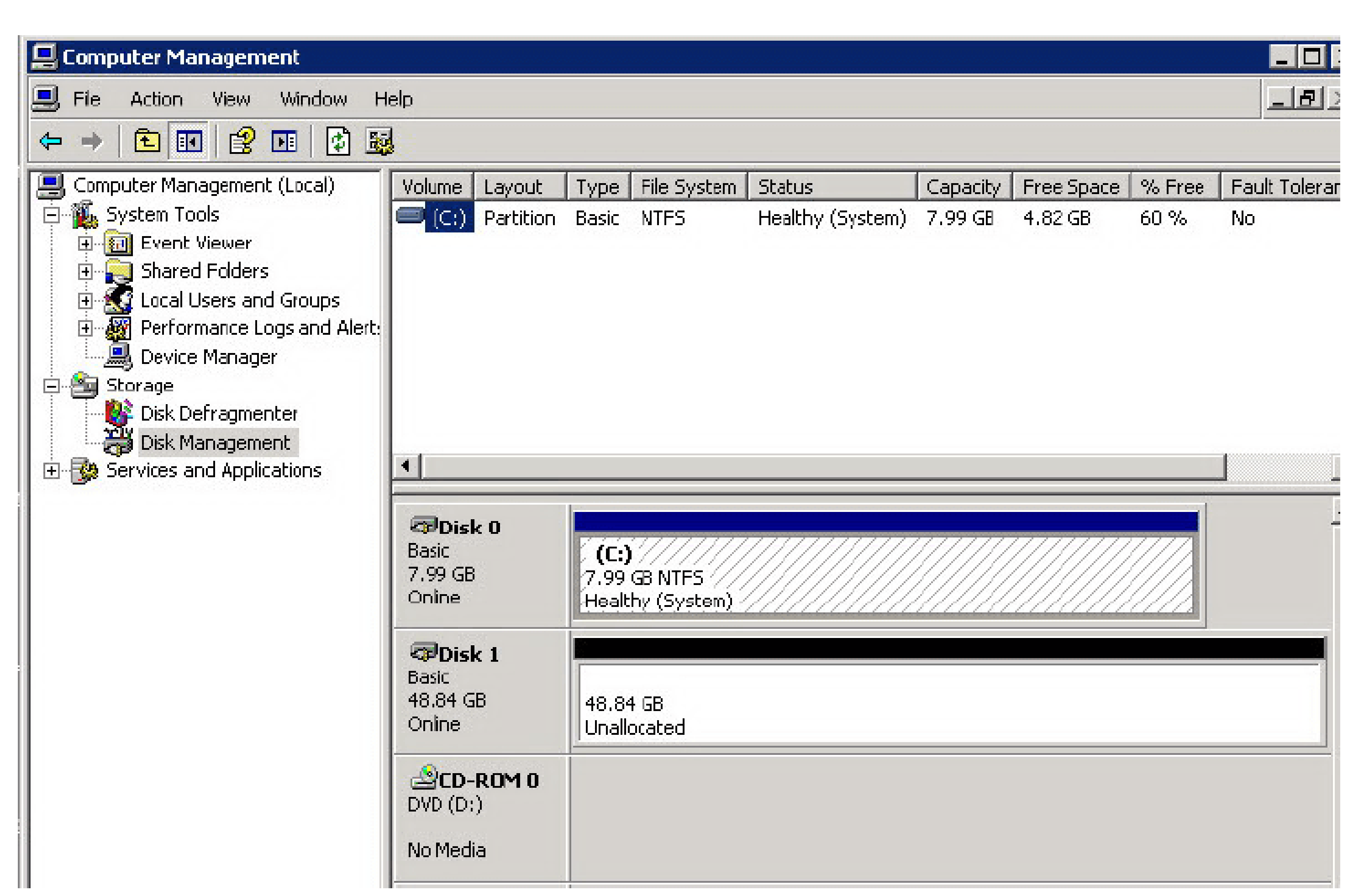
▶ 用 XP 观察 iSCSI 在操作系统下的对应

1．进入 XP 中，并且在我的计算机上右击，选择 Manage 选项。

2．选择 Disk Management。此时你会看到新增的磁盘，上面并没有安装任何文件系统。



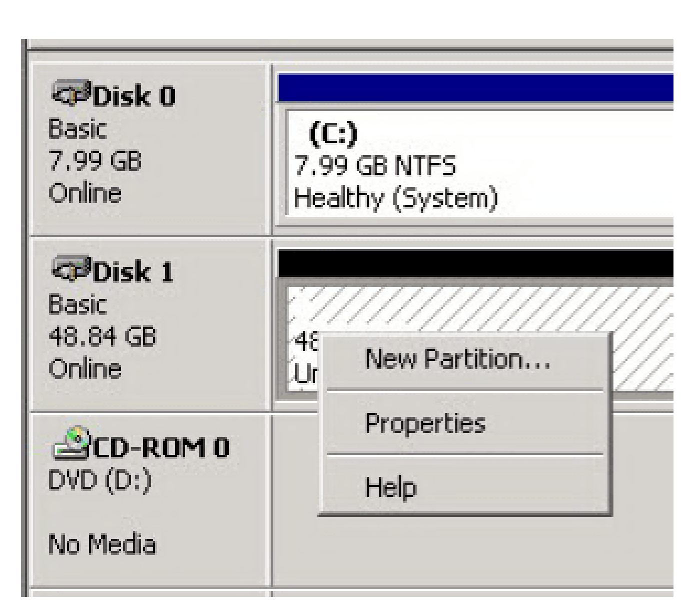
▲ 选择选项



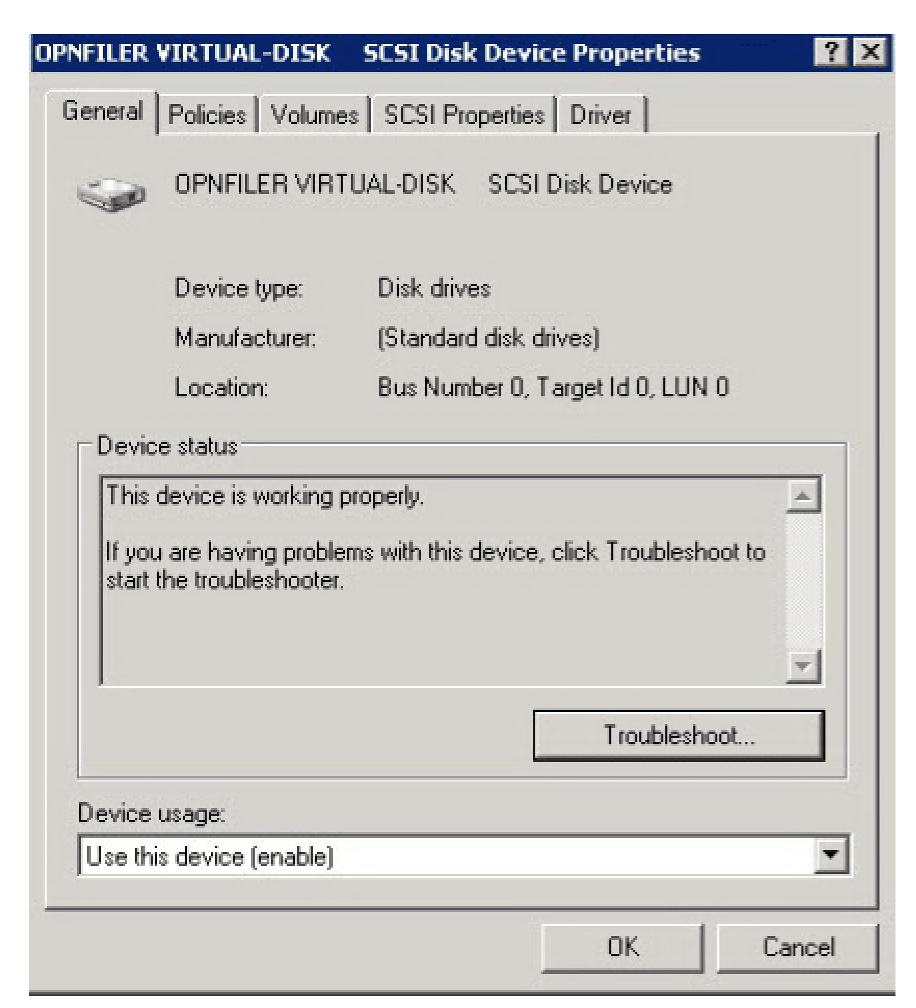
▲ 已经可以在这里看到 iSCSI 硬盘了

3．在这个磁盘上右击并选择 Properties 选项。

4．此时你可以看到这个磁盘，在 XP 下竟然是以一个单一分割区的方式呈现，让整个应用的灵活性大增。



▲ 选择选项



▲ 在 XP 下是一个分割区，可用灵活性大增

结 语

在这一章中，我们充分了解了 OpenFiler 的 iSCSI 配置以及连接，而不管是 VMware 或是 Hyper-V，都可以使用 iSCSI 的 LUN 来完成虚拟机的放置。有了独立的存储设备，整个虚拟机的功能才可能完整实现，因此在真正购买企业级的 SAN 或存储设备之前，使用免费的 OpenFiler 进行测试是最省成本的方案。甚至如果你使用的网络设备是 10GB 以上的交换机及网卡，更可以直接使用 OpenFiler 作为上线时的 VMware 存储设备，让整个虚拟机环境的成本又节省不少。

第 13 章 架设企业级的 vSphere 多服务器环境

关键词：

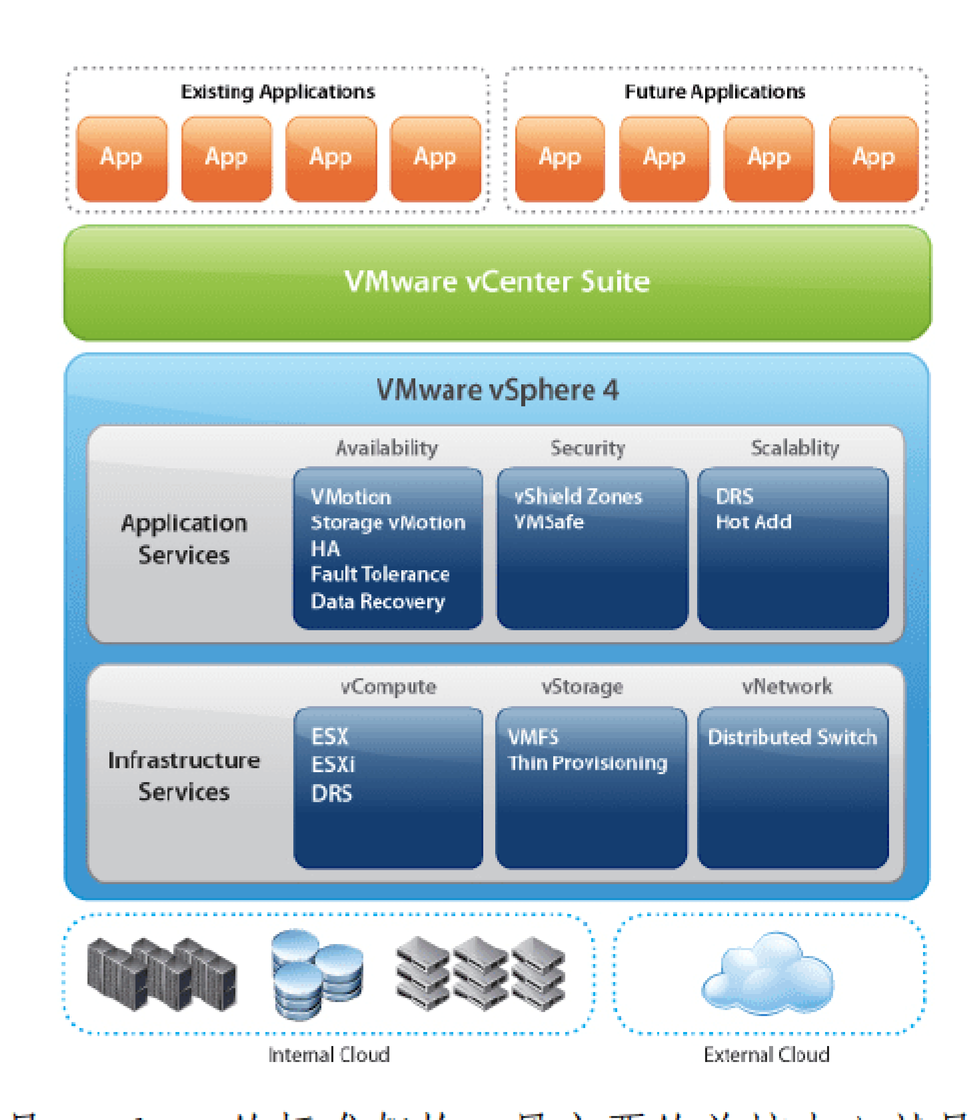
- 理解 vSphere 的架构，适应于多物理 ESX/ESXi 主机的场所
- 组成 vSphere 的 VMware 软件有 ESX/ESXi、vCenter 以及 vSphere Client
- 安装 vCenter 环境需要的是 Windows Server 系统以及 SQL 数据库
- 安装 vCenter
- 从 vSphere Client 连入 vCenter
- 理解 vCenter 的布局
- 理解 Datacenter、Cluster、Host 以及 VM 各个部件
- 理解各个部件的选项卡功能

ESX 的使用十分简单，充其量就是一个标准的 Hypervisor 来进行服务器的合并。然而 vSphere 的强大并不是靠单个 ESX 服务器就能体现出来。当 ESX/ESXi 组合成一个集群组合，再加上适当的管理界面，整个企业的 IT 架构将整个改观。vSphere 架构可以创建一个永远不会蓝屏，服务永远不中断，系统用户永远不会遇到问题，资源永远不会不够用的完美环境，这些听起来如天方夜谭的情节，vSphere 真的做到了！我们将在这一章中，介绍 vSphere 架构的组成部件、基本使用和各种高级功能（如 HA、DRS 和 VMotion）作准备。

13.1 理解 vSphere 架构

vSphere 是 VMware 在企业应用的旗舰产品。严格来说，vSphere 并不是一个单个产品，而是由 VMware 中不同的产品所组合而成的解决专案。vSphere 架构的目的并不是单纯的虚拟化，而是提供企业一个完整的平台，让整个企业的业务得以延续，不会因为任何 IT 本身的软硬件或人为操作疑义而停止。

通常在企业中，最担心发生的事的就是服务中断，包括用户访问服务器的服务（如邮箱）、用户本身的桌面服务（如 Office 应用）等。这些服务的停止通常会给企业造成严重的损失，因此 vSphere 的目的，就是使用虚拟机技术，让所有的服务能在停止时能立马回退，或是让用户根本在还没有察觉服务已经中断之前，能自动将服务恢复了。



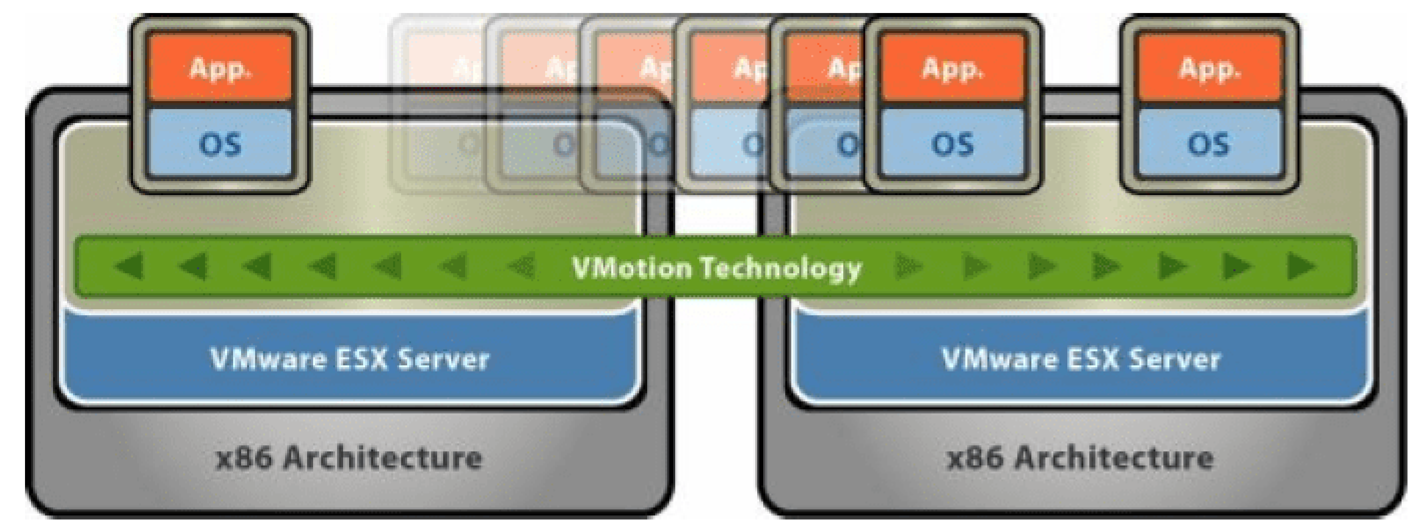
▲ 这是 vSphere 的标准架构，最主要的总控中心就是 vCenter

13.1.1 vSphere 解决专案适用的环境及原理

由于虚拟机拥有强大的备援和灾难回退的速度，因此会让系统的停机时间减少到微秒以下。因此我们在这里要知道，vSphere 并不是一个虚拟机的产品，而是一个利用虚拟机技术的企业整体解决专案。而要成为一个完整的企业解决专案，只靠虚拟机是不够的，因此在一个完整的 vSphere 架构中，需要不同的软硬件来配合。

1. vSphere 最适用的环境：多服务器的环境

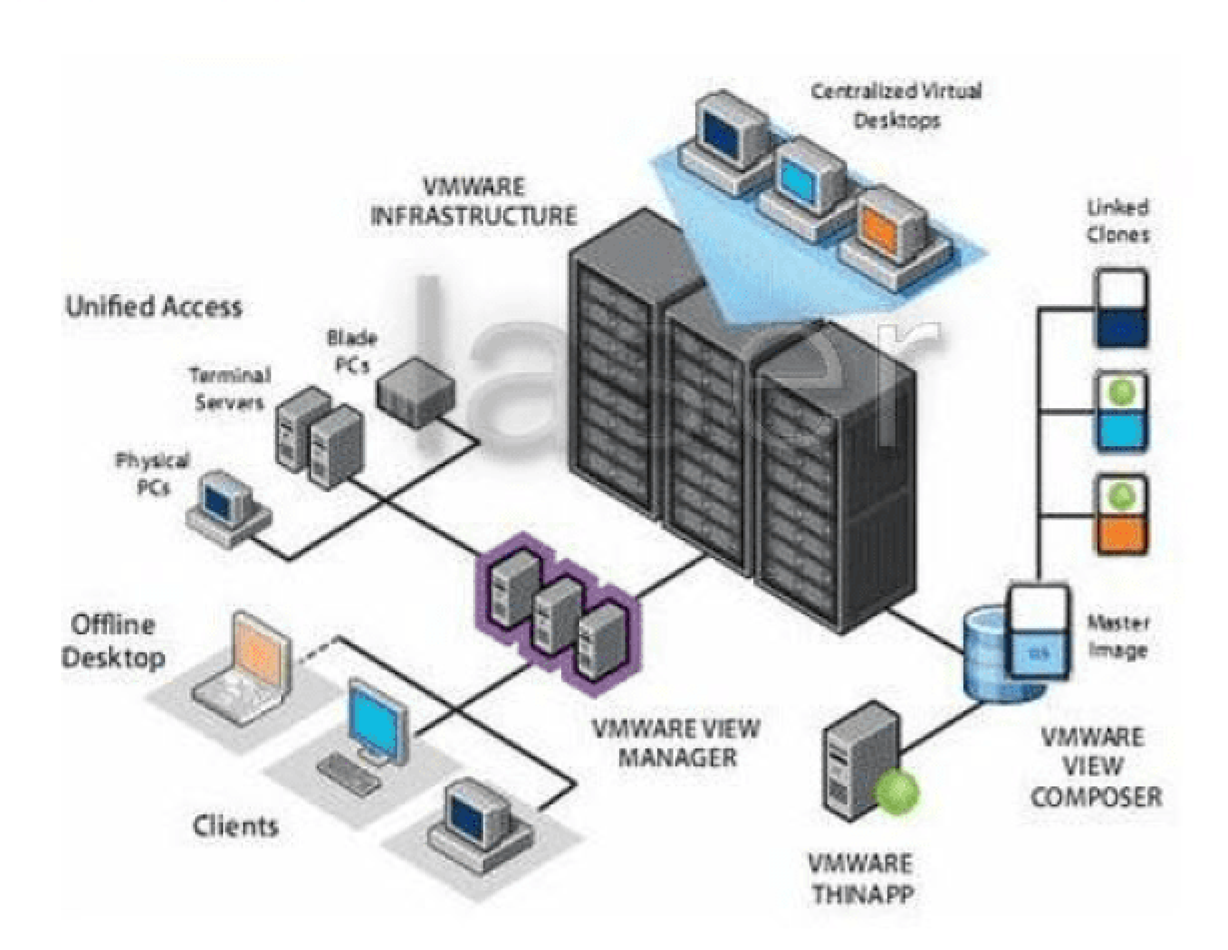
vSphere 主要分成两个方面，最主要的还是应用在服务器的集成。大部分的企业服务器都使用了集群来进行备援以达到永不停机的目的。但多台虚拟机放在一台 ESX 的物理机上还是会有危险，因此 vSphere 架构的目的，不但提供多台 ESX 形成的集群，也会利用一些管理的模块来落实 ESX 物理机上虚拟机的转移（如 HA、DRS 或 Vmotion）。因此第一个适用 vSphere 的环境就是所有拥有服务器的企业，换句话说，全世界的所有企业都是 vSphere 架构的潜在用户，而 vSphere 架构在企业的服务器环境中，最实用的功能就是提供永不停机的服务。



▲ 在多物理机的环境中，可以让 VM 在 ESX/ESXi 之间移动

2. 虚拟桌面的环境

当企业的客户端都改成了 Thin Client 或是 Zero Client 之后，所有客户端的 OS 都全部安装到服务器上的虚拟机中。这时所有的 OS 都被集中在 vSphere 架构的管理中，不管是系统升级，灾难回退或是实用程序的安装，都可以用中央控管，因此 vSphere 架构在虚拟桌面的功能上，最佳的应用就是提供最方便的管理机制。



▲ 虚拟桌面是当前虚拟化的新星，产品是 VMware View

3. vSphere 背后的技术

vSphere 背后的技术当然就是虚拟机、ESX 物理服务器群集和独立的存储系统。ESX 物理服务器上安装 Hypervisor 提供虚拟机的平台，独立存储使用 RAID 提供了永远不会消失的数据存放，而集群更让 ESX 的物理机不会因系统故障而无法提供 Hypervisor，当然这一切的组合，就是为了要让所有的服务都运行在虚拟机中，读者们可以回到第 1、2 章，多理解 vSphere 的技术及部署原理。



▲ 主要技术是以 ESX/ESXi 为中心的 VM

13.1.2 组成 vSphere 的部件

vSphere 是由固化的软硬件和服务组成的。在硬件方面，我们已经在本书的前几章说明了。包括匹配 HCL 的服务器、存储设备、网络设备、FC SAN 交换机等。读者们可以回到本书的前五章复习。

1. vSphere 架构需要的 VMware 软件

在软件方面，vSphere 需要的最基本的就是 ESX/ESXi Server 以及客户端 vSphere Client。然而 vSphere 管理的不是单纯的一台 ESX/ESXi，而是由 ESX/ESXi 形成的集群，因此必须有一个管理的机制，因此在 vSphere 架构下，会比单独的 ESX 管理多出新的成员，下面就是成员的列表。

▶ vSphere 架构下需要的服务器

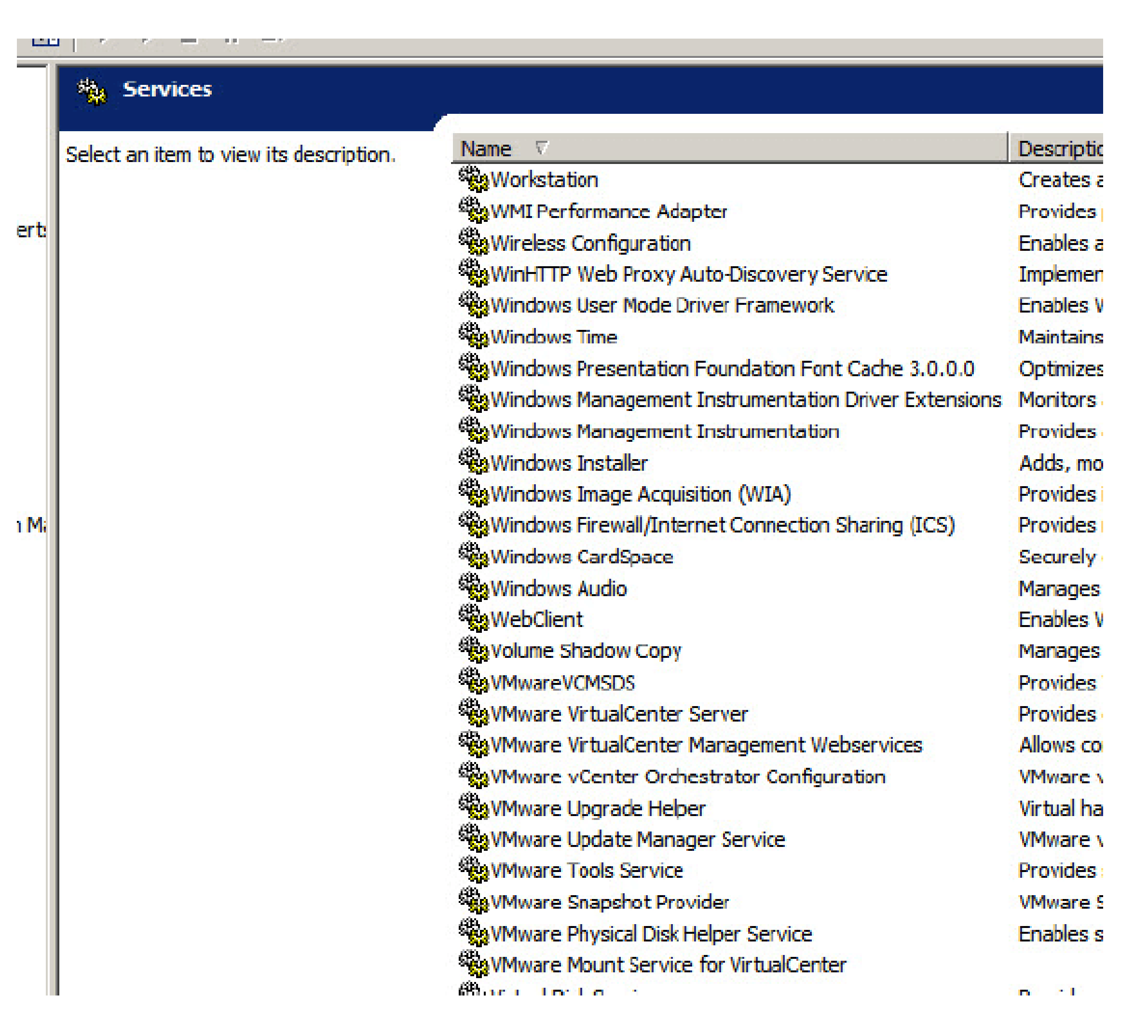
（1）安装 ESX/ESXi 的物理服务器（并取得用户许可证或试用授权）。

（2）安装 VMware vCenter 的物理服务器（使用 Windows Server）。

（3）管理 vSphere 的 vSphere Client 客户端，但并不是连入 ESX，而是连入 VMware vCenter。

```
[root@esxvm ~]#
[root@esxvm ~]# uname -a
Linux esxvm 2.6.18-128.ESX #1 Thu Oct 15 16:11:16 PDT 2009 x86_64 x86_64 x86_64
GNU/Linux
[root@esxvm ~]# _
```

▲ 使用 ESX/ESXi 的物理机是基本的



▲ 加上 vCenter　　　▲ 使用 vSphere Client 连入 vCenter

2. vSphere 架构需要的操作环境

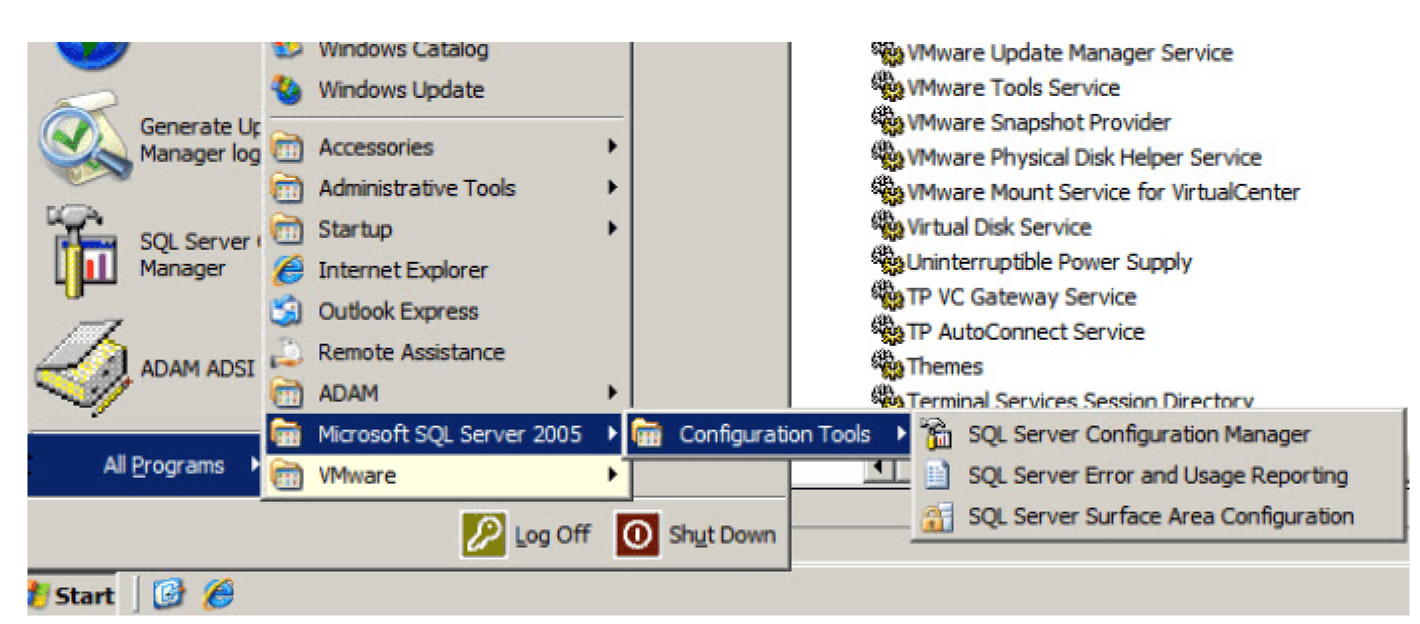
大部分的计算机用户都是使用 Windows 系统，而 vSphere 架构除了 ESX 本身和存储设备自身的操作系统之外，其他的软件都是运行在 Windows Server 之下。新版的 vSphere 4.0 不再需要使用 Windows 的主动目录环境，可以在任何一台 Windows Server 上安装。此外 VMware vCenter 要管理多台 ESX/ESXi 服务器，因此需要数据库来存放。

▶ vSphere 架构下需要的基础环境

（1）数据库系统：在安装 vCenter 时，会附简单的 SQL 2005 简易版。

（2）安装 VMware vCenter 的 Windows Server。

（3）vSphere Client 客户端机器（使用 Windows 系统）。

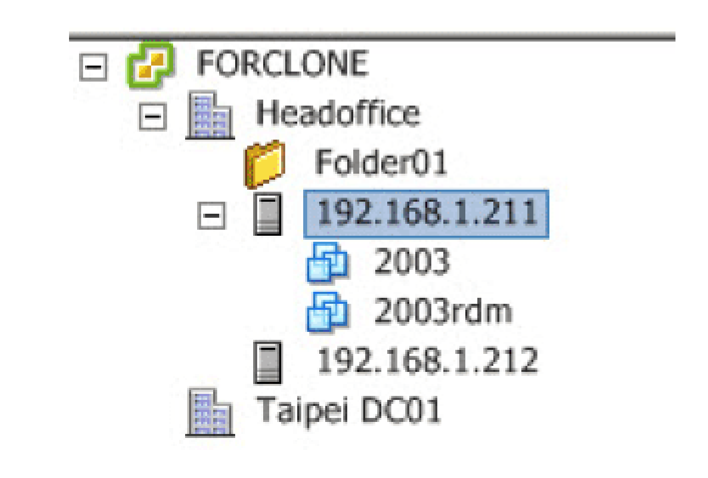


▲ 新版的 vCenter 不需要 AD 也可以使用，也附了简易的 SQL

事实上，一个最简单的 vSphere 系统所需要的是两台 ESX 服务器、一台安装 Windows Server 的计算机（可使用虚拟机），这台 Windows Server 上必须安装 SQL。

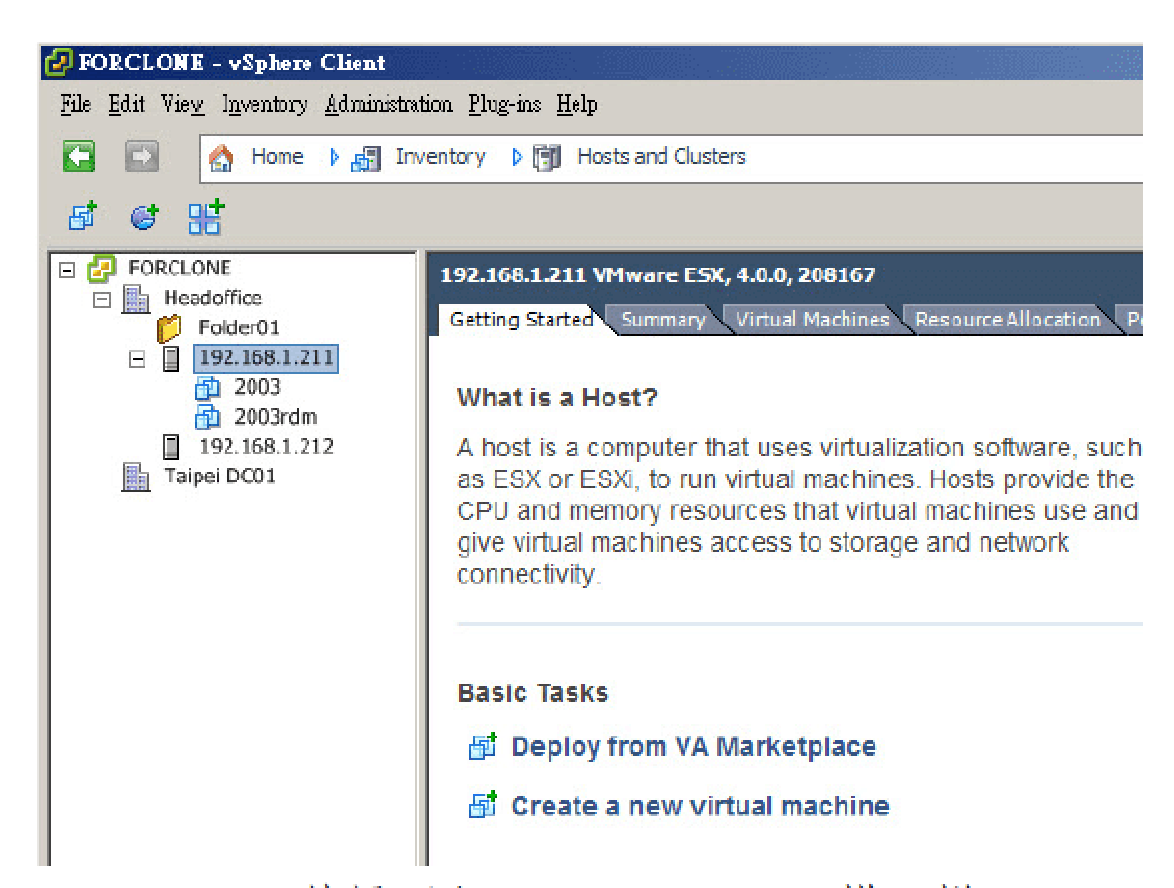
3. VMware vCenter 简介

vCenter 是 vSphere 中最重要部件，在前面几章讲过有关 ESX 的操作，所有的功能都是在独立的 ESX 和 vSphere Client 之间管理和操作。但 vSphere 架构是一个多 ESX 主机的环境，vSphere Client 已经无法管理多台 ESX 服务器，更遑论将多台 ESX 集成应用。此时我们需要一个类似 MMC 的界面，将多台 ESX 纳入管理，并且加入用户的权限管理、多台 ESX 服务器的数据导出导入。这就是 VMware vCenter 的功能所在。



▲ VMware vCenter 最重要的功能就是管理多台 ESX

在 vSphere 架构中，vCenter 除了以上的功能之外，更提供了 VMware 专用的功能界面，如 HA、DRS、VMotion 等操作环境，因此在整个 vSphere 架构中，vCenter 可说是真正的灵魂。vCenter 在 vSphere 架构中包括了 VMware vCenter，用来管理所有的 ESX/ESXi 虚拟机，另外也有我们已经熟悉的 vSphere Client，主要是连入 VMware vCenter 中的客户端，在熟悉了 vCenter 的 Server 和 Client 之后，vSphere 的使用大致就明白了。



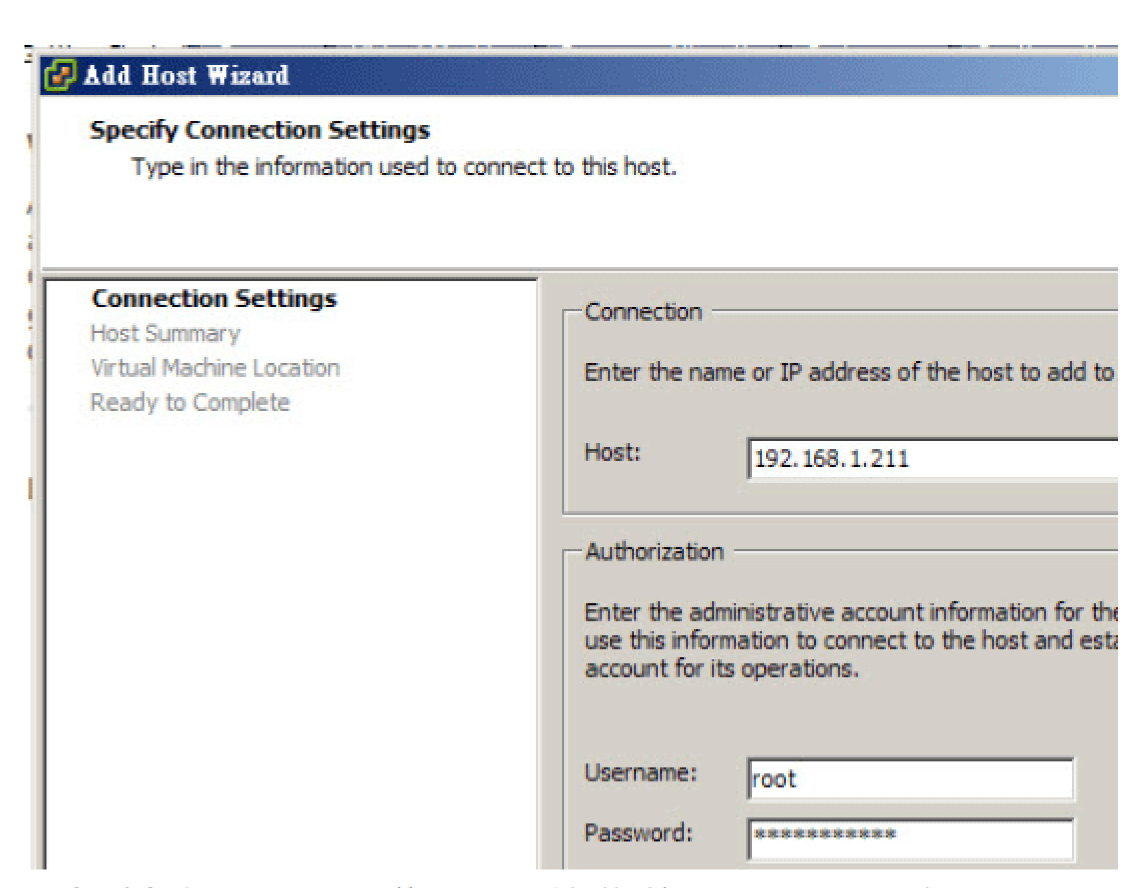
▲ 其界面和 vSphere Client 一模一样

13.2 准备 vSphere 的安装环境

安装 vSphere 部件分成三部分，首先是将 ESX 服务器的主机名配置完毕，并且遍历 ESX 是否拥有 HA、VMotion 等高级功能的授权。配置完之后，就需要将 Windows 2003 Server 的环境架设起来，并且将 SQL 安装起来，之后才是安装 vCenter，在落实 VMware vCenter 安装后，就是激活 VMware vCenter 了，并且从客户端来访问 VMware vCenter。我们在这一节，就先来看看安装基本的环境配置。

13.2.1 配置 ESX 服务器

创建 vSphere 的第一步就是 ESX 的物理服务器创建。ESX 服务器的安装和创建虚拟机可以引用前面几章的详细说明。在本章的示例中，我们必须将两台 ESX 服务器加入 vCenter 中。在新版的 vSphere 中，VMware 不再强调一定要使用 Windows 主动目录及 DNS 了。只要有 ESX 的 IP 即可，让整个 vSphere 的配置更加简单化了，我们只要确定要加入 vCenter 的 ESX 主机 IP 即可。



▲ 新版的 vSphere 使用 IP 就能将 ESX/ESXi 加入 vSphere

1. 确定 ESX 主机 IP

在安装 ESX/ESXi 时，别忘了记下每一台主机的 IP。你也可以使用主机名以及 DNS 来管理，但整个架构会较为复杂。

2. 遍历 ESX 的授权

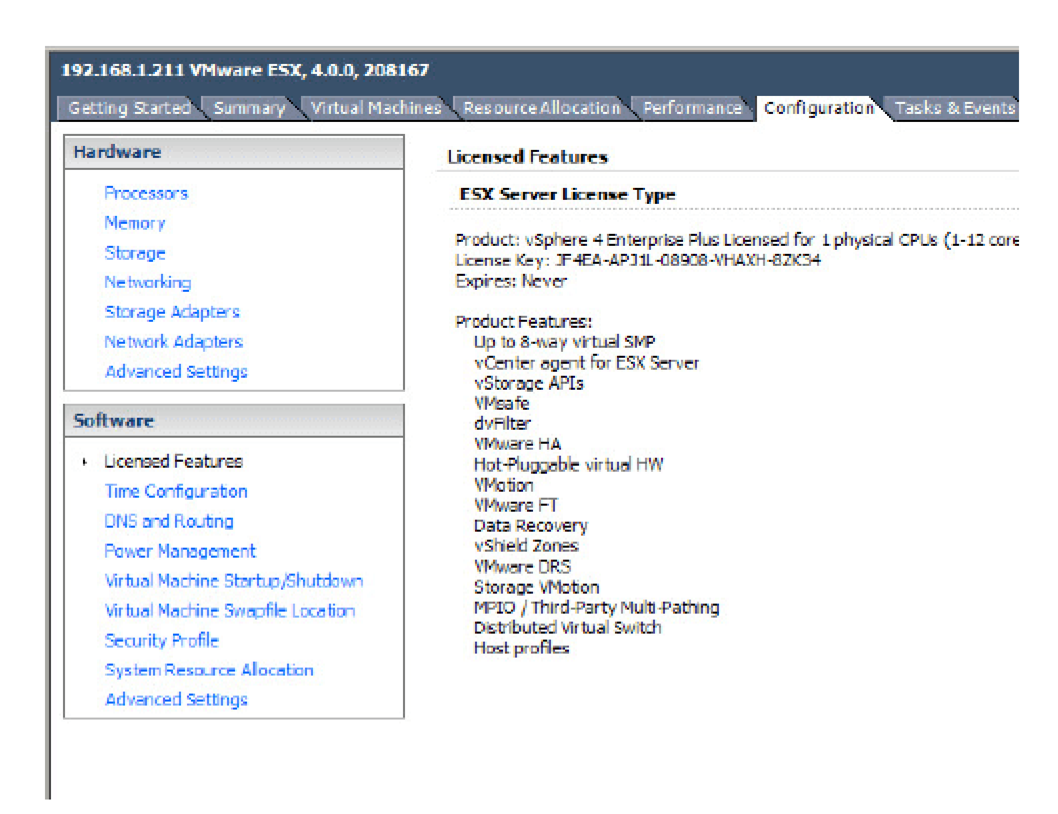
如果你是 vSphere 的试用者，可以到 VMware 的官方网站申请试用的授权，才能使用 HA、DRS、VMotion 等高级功能。如果你是合法的授权用户，可以用下列的方式加入授权，或是遍历你的 ESX 是否有授权。

▶ 遍历 ESX 的授权

（1）进入 vSphere Client，并且选择 Configuration 中的 Licensed Features。

（2）进入之后可以看到当前这台 ESX 的授权情况。

（3）ESX 服务器支持几种授权方式，当前没有安装的功能必须要等到虚拟机安装上才会激活。



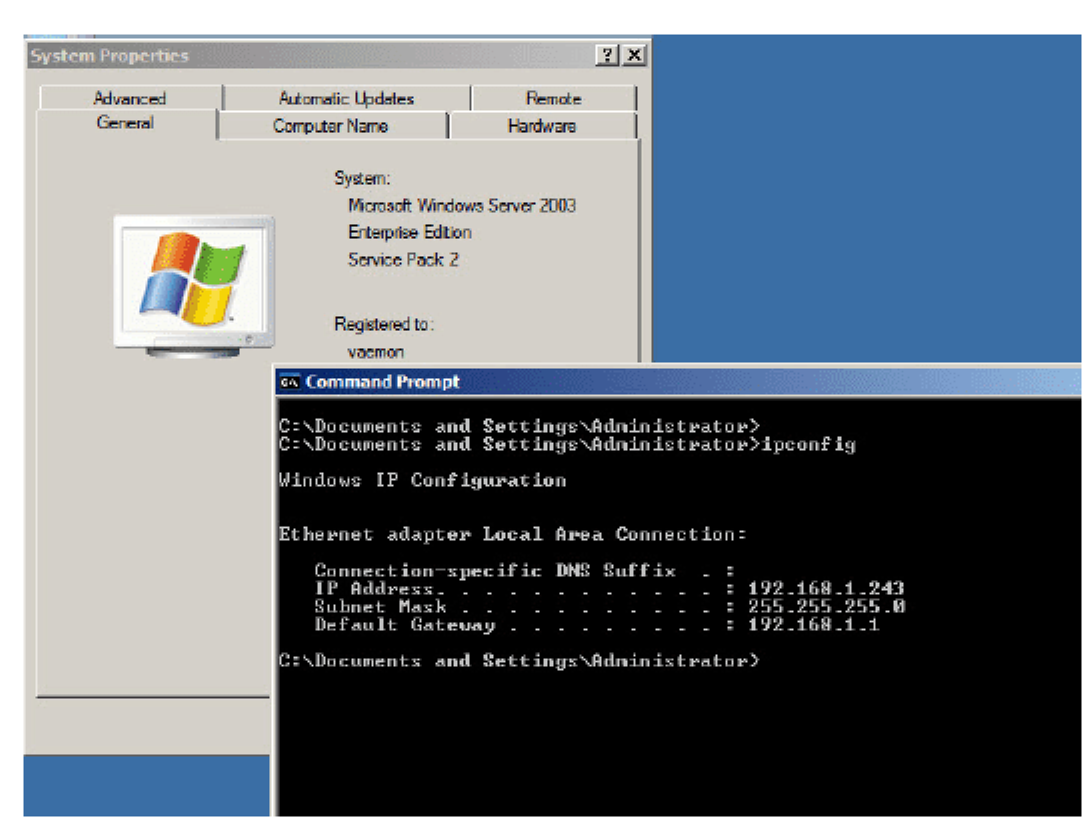
▲ 检视 VMware 的高级功能授权

13.2.2 安装基本的 Windows 2003 Server 环境

vSphere 需要一个 Windows 2003/2008 Server 主机，并且安装好 SQL 即可。在 vSphere 中，SQL 已内置在 VMware vCenter 中，因此这里只要安装好 Windows 2003 Server 即可。

▶ 安装 Windows 2003

首先我们在一台物理机上安装 Windows 2003 Server，并且将这台 Windows 2003 补丁到最新的 Service Pack。特别要注意的，这台 Windows 2003 必须拥有 ESX 主机的同一网段，本书大部分的示例将使用 192.168.1.X 的网段。



▲ 这是在物理机上的 IP 配置

13.3 安装及 VMware vCenter 简介

新版的 vSphere 安装 vCenter 的步骤及使用都整个简化了，你只要拥有一台 Windows 2003/2008 即可。而安装 vCenter 的步骤就如同安装任何一个 Windows 下的实用程序一样简单，我们就来看看安装的步骤，并且在安装完之后，大致理解一下 vCenter 的里面及基本使用。

13.3.1 安装 VMware vCenter

在安装好 Windows 2003 之后，就该安装 VMware vCenter 了。在安装好 vCenter 之后，我们可以在这台 Windows 2003 上直接安装使用 vSphere Client，虽然可以省下一台机器，但在生产环

境上是不建议这么做的。另外在安装好 vCenter 之后，也可以顺便安装 Update Manager，以方便 ESX/ESXi 以及 vCenter 本身的补丁。

1. 安装 VMware vCenter

安装 vCenter 的步骤十分简单，主要就是将 vCenter 安装至计算机，并且安装 SQL 服务，最后再以服务模式来激活 vCenter，以方便 vSphere Client 的连入。

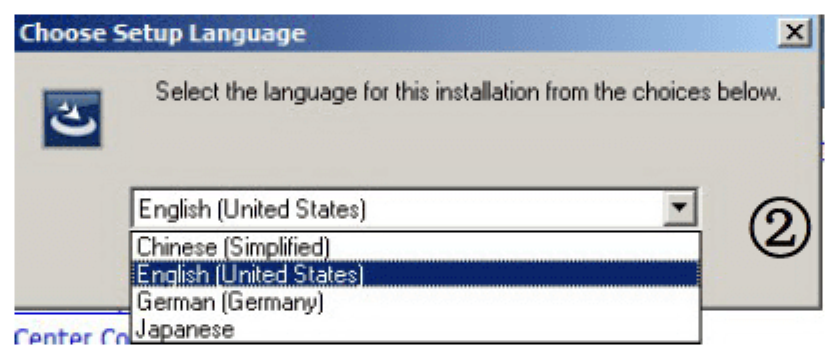
> **注意**
> **将 VMware vCenter 安装在 VM 中**
> 有趣的是，你可以将 VMware vCenter 安装在需要管理的 ESX/ESXi VM 中，让这个 VM 中的 Windows 2003 来管理其宿主系统，虽然这样可以省下一台机器，但许多功能就无法落实，而如果你这个 vCenter 所在的 ESX/ESXi 重启，你就失去总控 VMware vCenter 的能力了。

▶ 安装 VMware vCente

1. 首先将 VMware vCenter 的光盘片放入光盘中，在自动运行激活后，会看到如下图所示的画面，选择 vCenter server 安装。
2. 此时会进入安装画面，选择第二项 English。

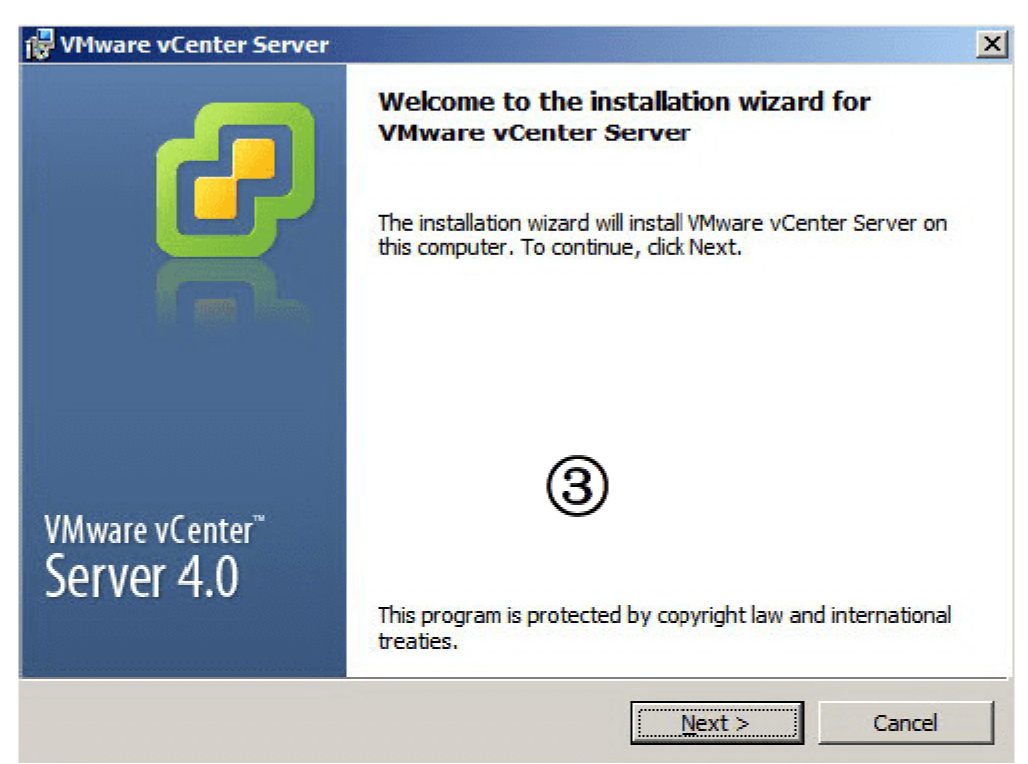


▲ 选择 vCenter Server 安装

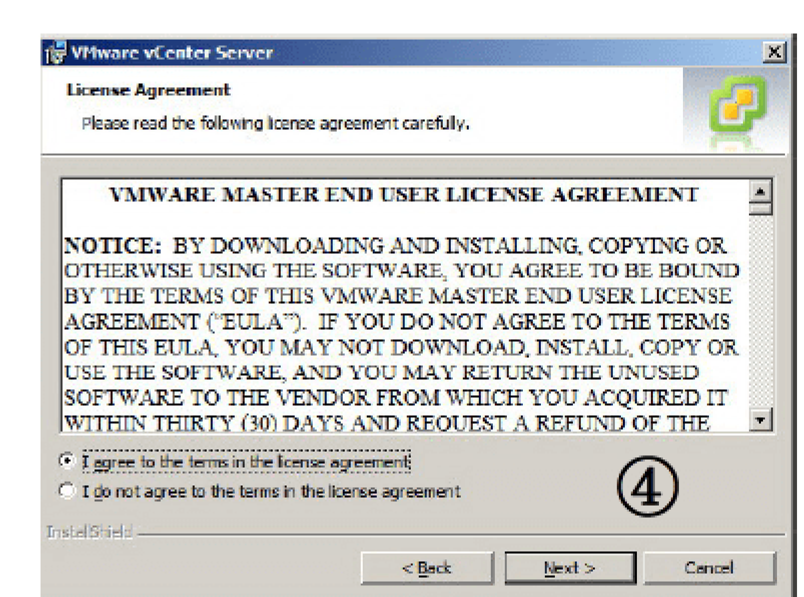


▲ 选择安装语言

3. 接下来会进入欢迎画面，单击 Next 按钮继续。
4. 接下来是授权协议，选择同意之后，单击 Next 按钮继续。

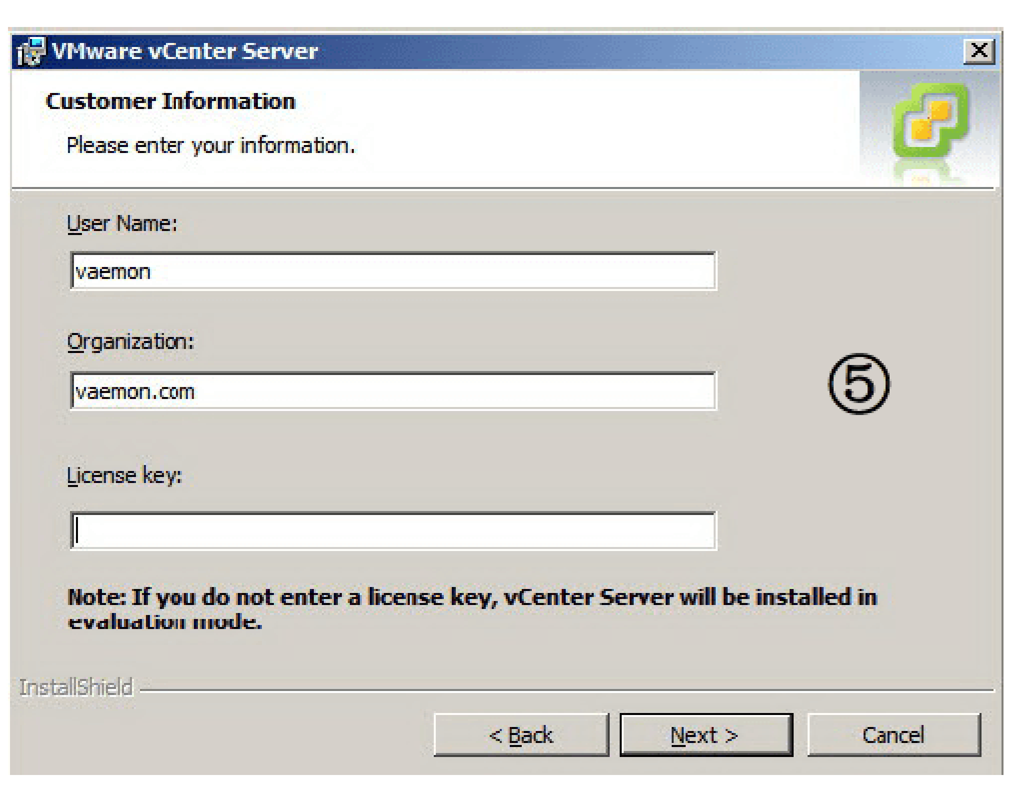


▲ 欢迎画面

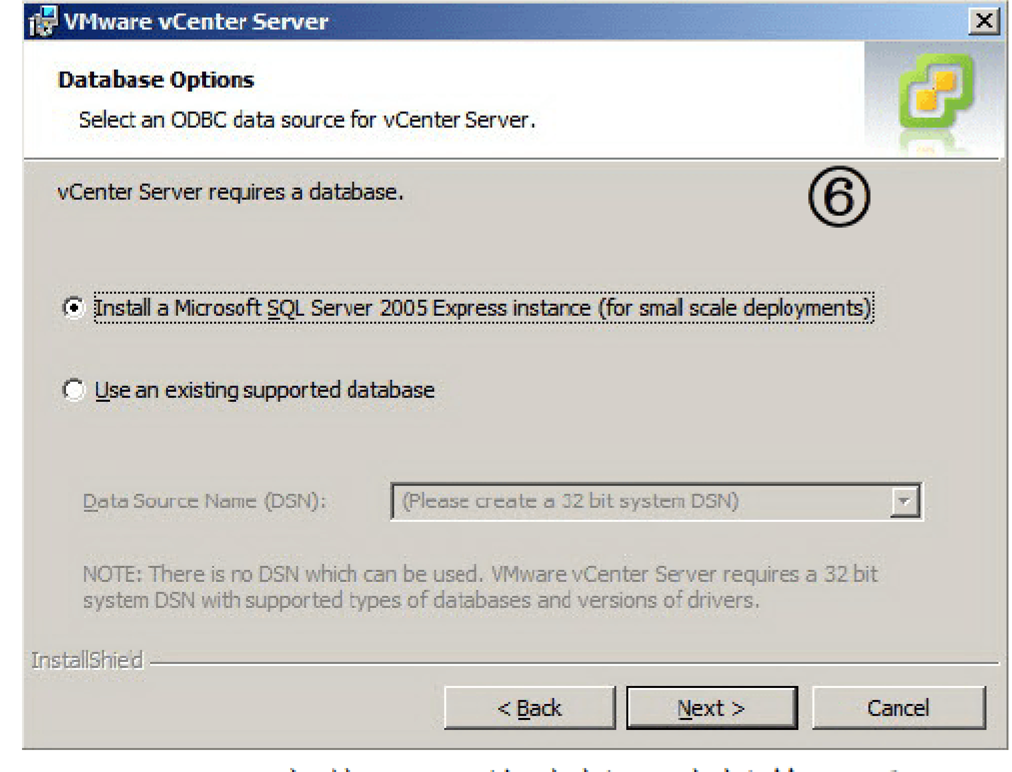


▲ 授权协议

5. 接下来是键入授权的序号，如果你有购买 vCenter，则键入序号，你也可以跳过序号键入，以试用版的方式进行，单击 Next 按钮继续。
6. 接下来是安装 SQL，我们这台 Windows 2003 当前还没有安装 SQL，因此就选择第一项，单击 Next 按钮继续。

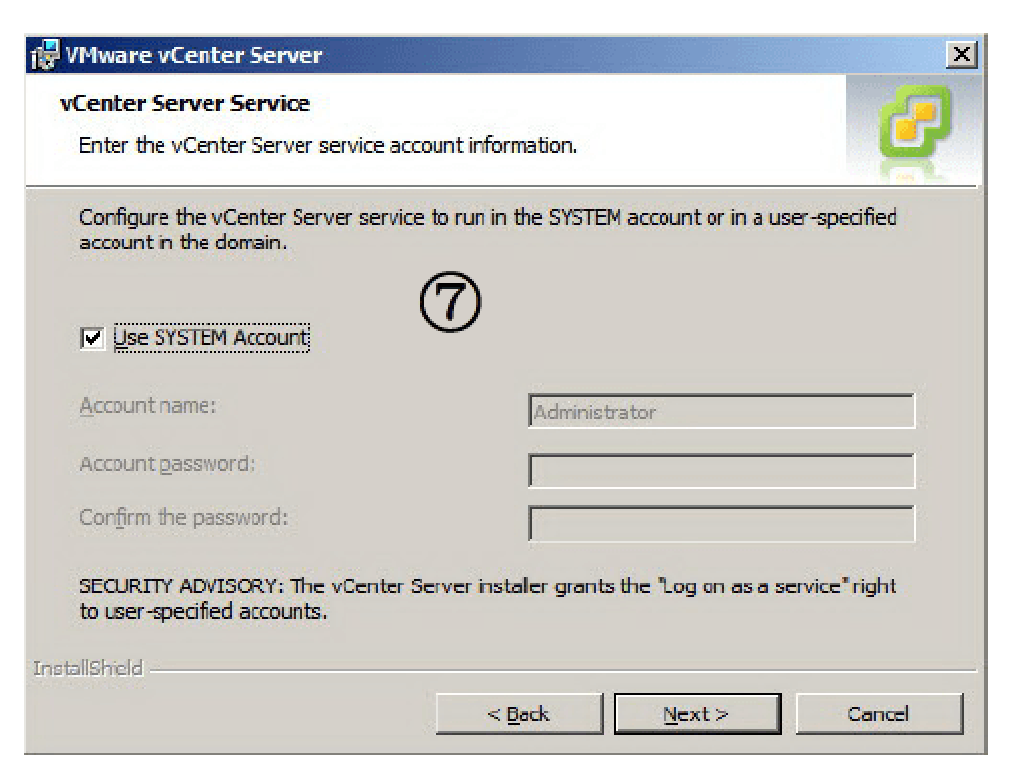


▲ 键入序号

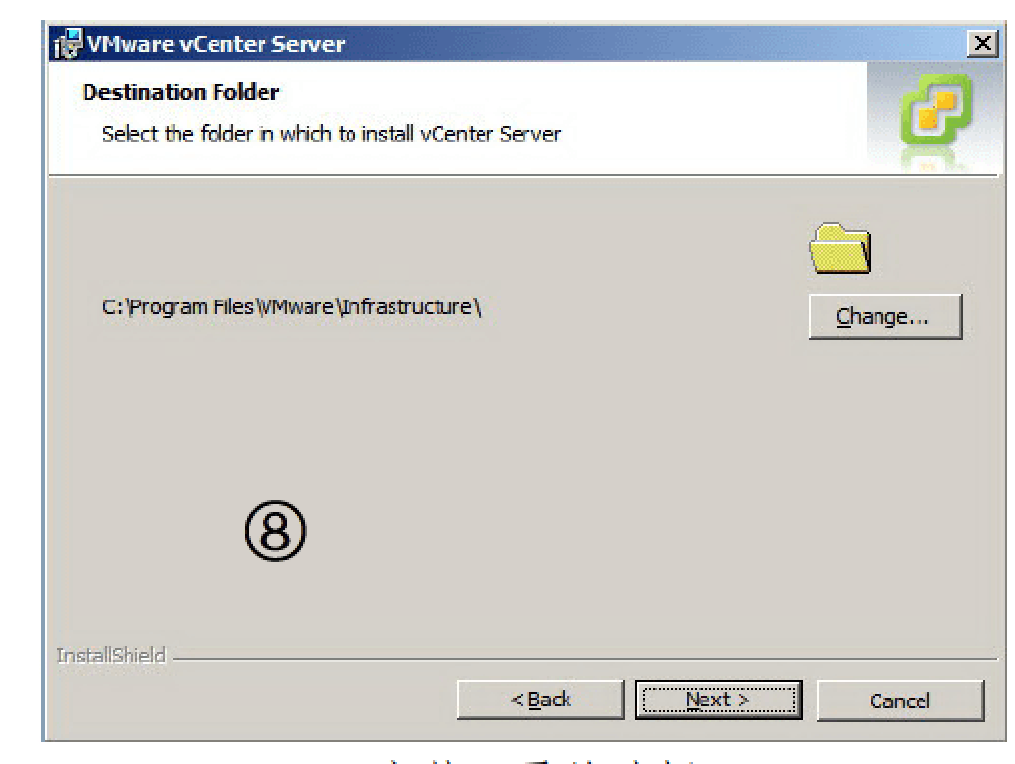


▲ 安装 SQL 的选择，选择第一项

7. 这是 vCenter 的激活模式，我们可以在这里键入 vCenter 的账号口令，也可以直接使用 Windows 的账号口令来作为登录的账号，本例使用 Windows 的账号，单击 Next 按钮继续。
8. 接下来是安装目录，本例以默认值为主，单击 Next 按钮继续。

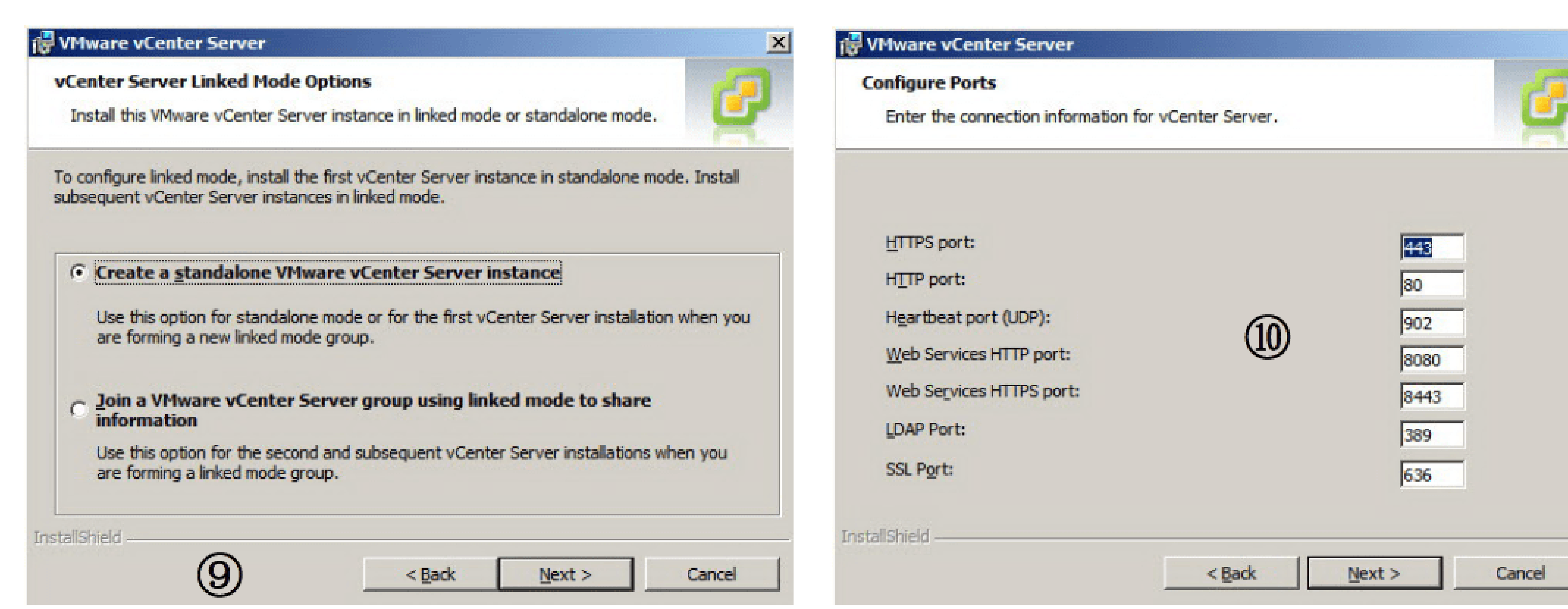


▲ 使用系统账号在 Windows AD 网域内可以和域账户集成，较为方便



▲ 安装目录的选择

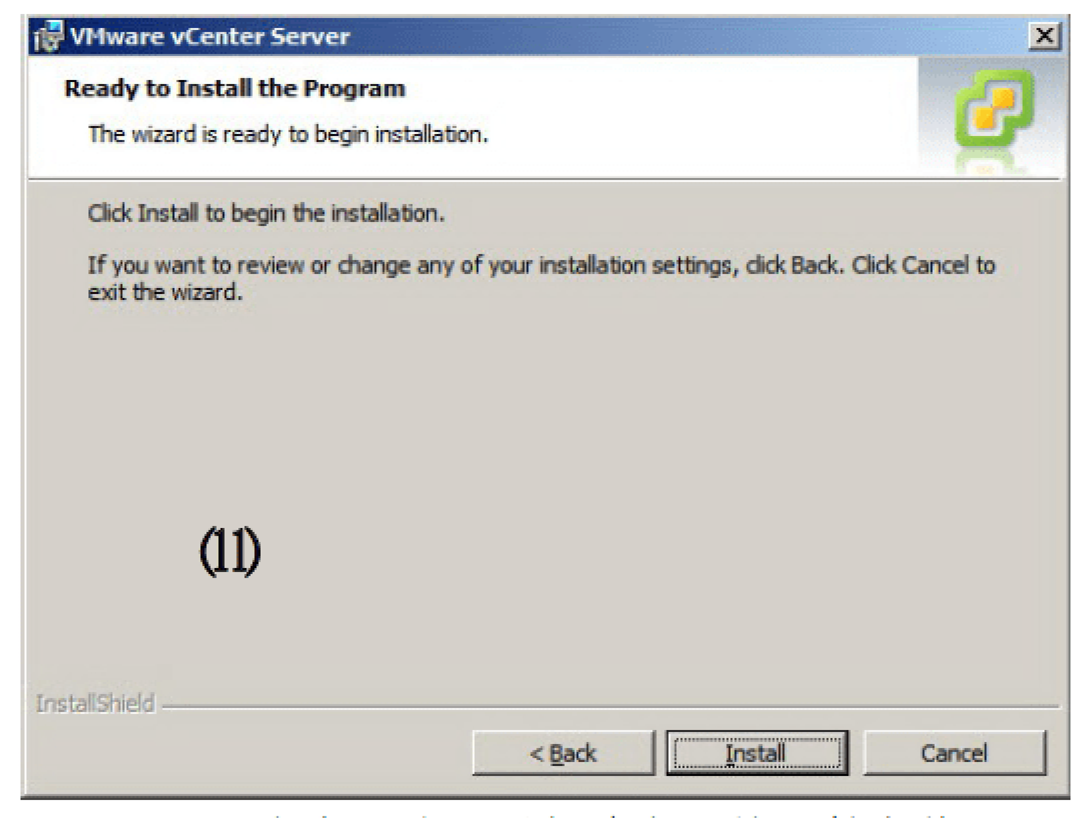
9. 接下来是安装模式。一般来说，你可以在多个 ESX/ESXi 中创建多个 VMware vCenter 的主机，称为 Linked Mode。本例是第一台，因此选择第一项的单台即可，单击 Next 按钮继续。
10. 接下来是连入 vCenter 的通信端口配置，我们就使用默认值，单击 Next 按钮继续。



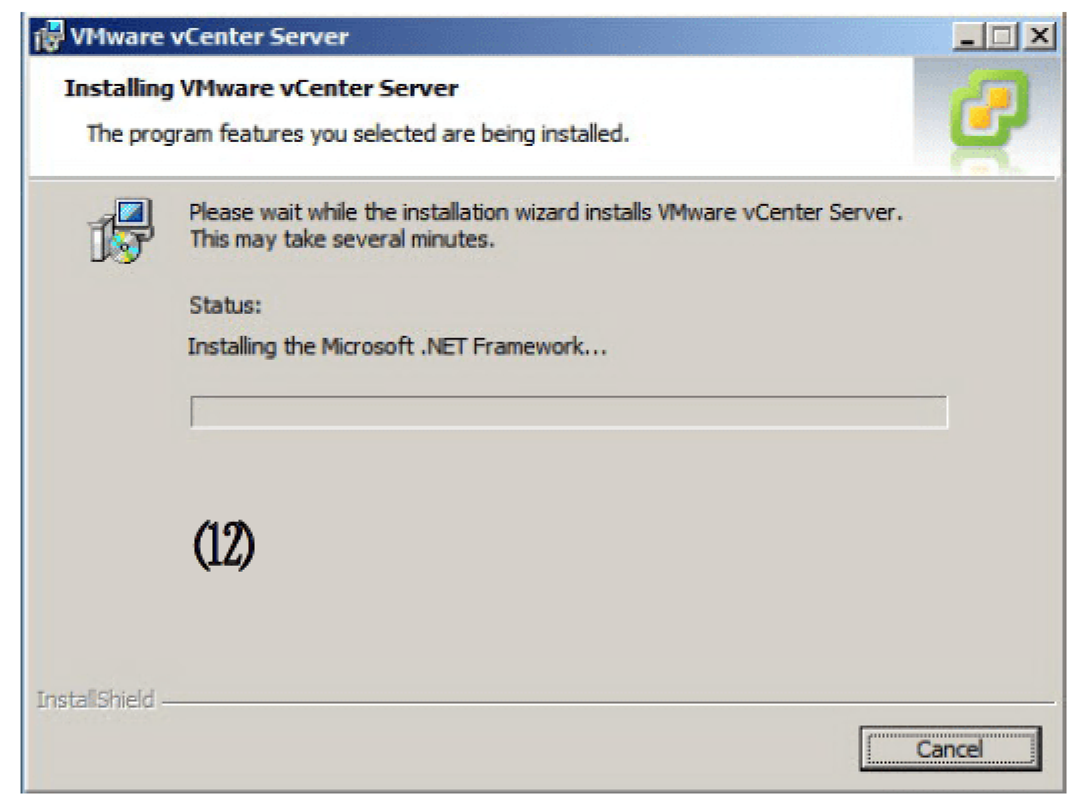
▲ 这是第一台 vCenter，因此我们就选择第一项单独安装　　▲ 一般通信端口的配置不需要更动

11．接下来是正规安装的开始了，单击 Install 按钮。

12．此时会进入安装画面，系统会先安装 Microsoft .NET Framework。



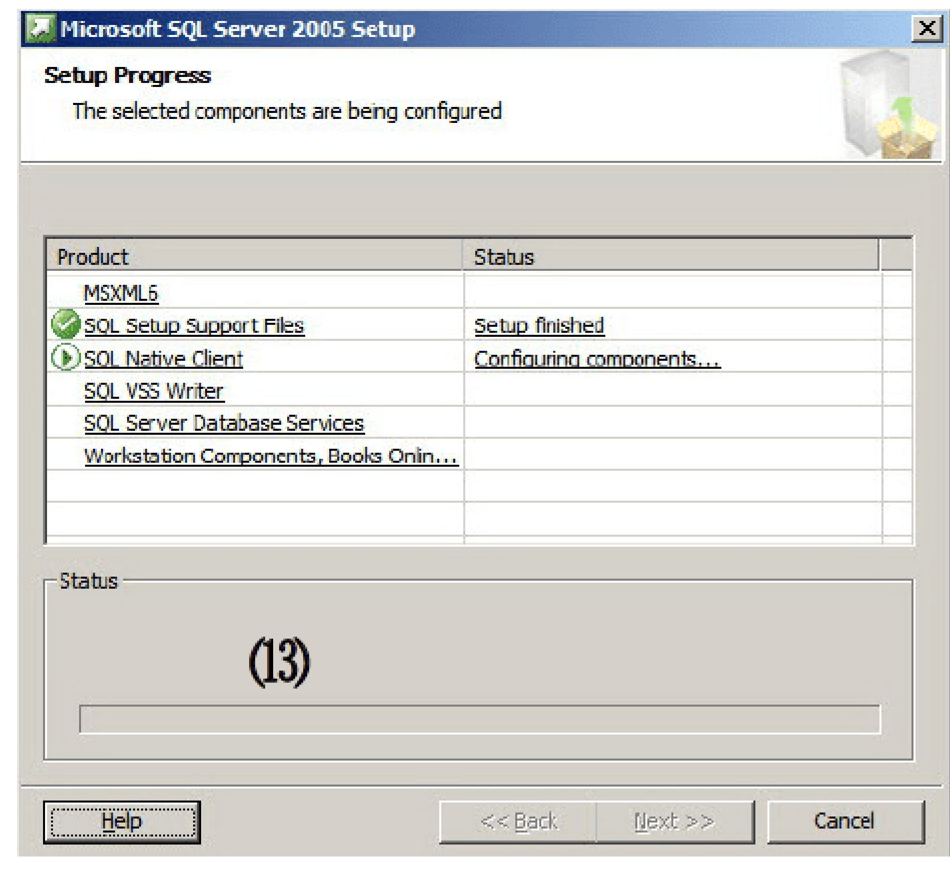
▲ 当上面的配置都确定后就开始安装



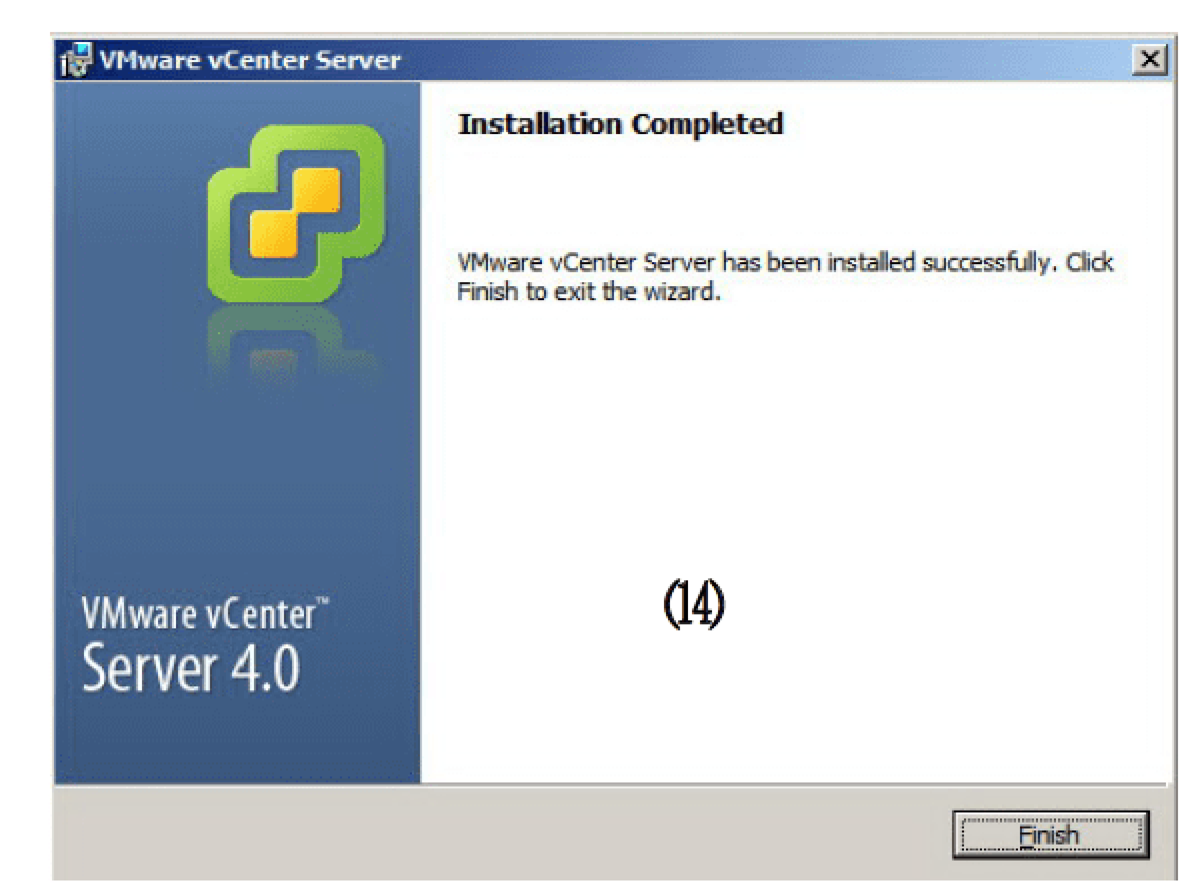
▲ 开始安装

13．在安装过程中，系统会进行简易版 SQL 2005 的安装。

14．当弹出如下图所示的画面时，意味着 VMware vCenter 已经安装完毕了。



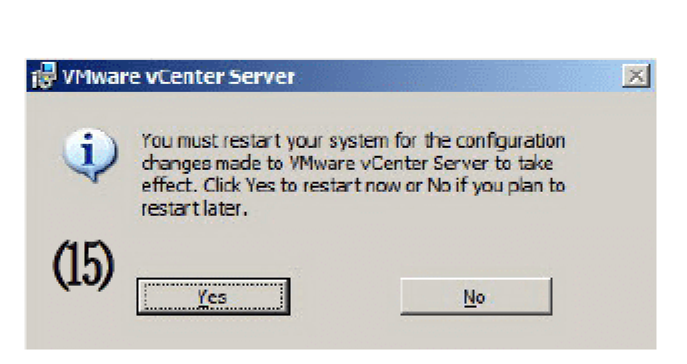
▲ 安装 SQL Server

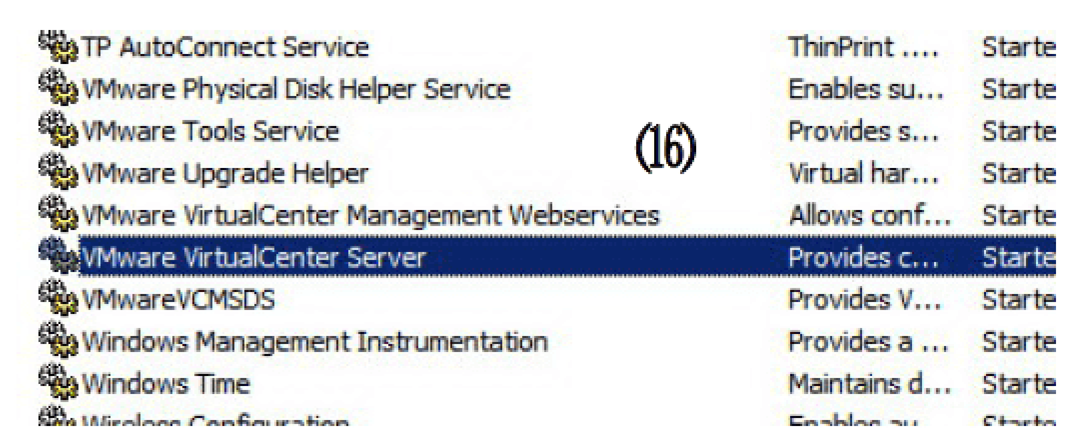


▲ 安装完毕

15．由于 vCenter 是服务，因此必须重新激活。

16．已重新激活，事实上 vCenter 已经以服务方式激活了。





▲ 必须重新激活，vCenter 在 Windows 中是一个服务　　▲ 这个服务已经激活了

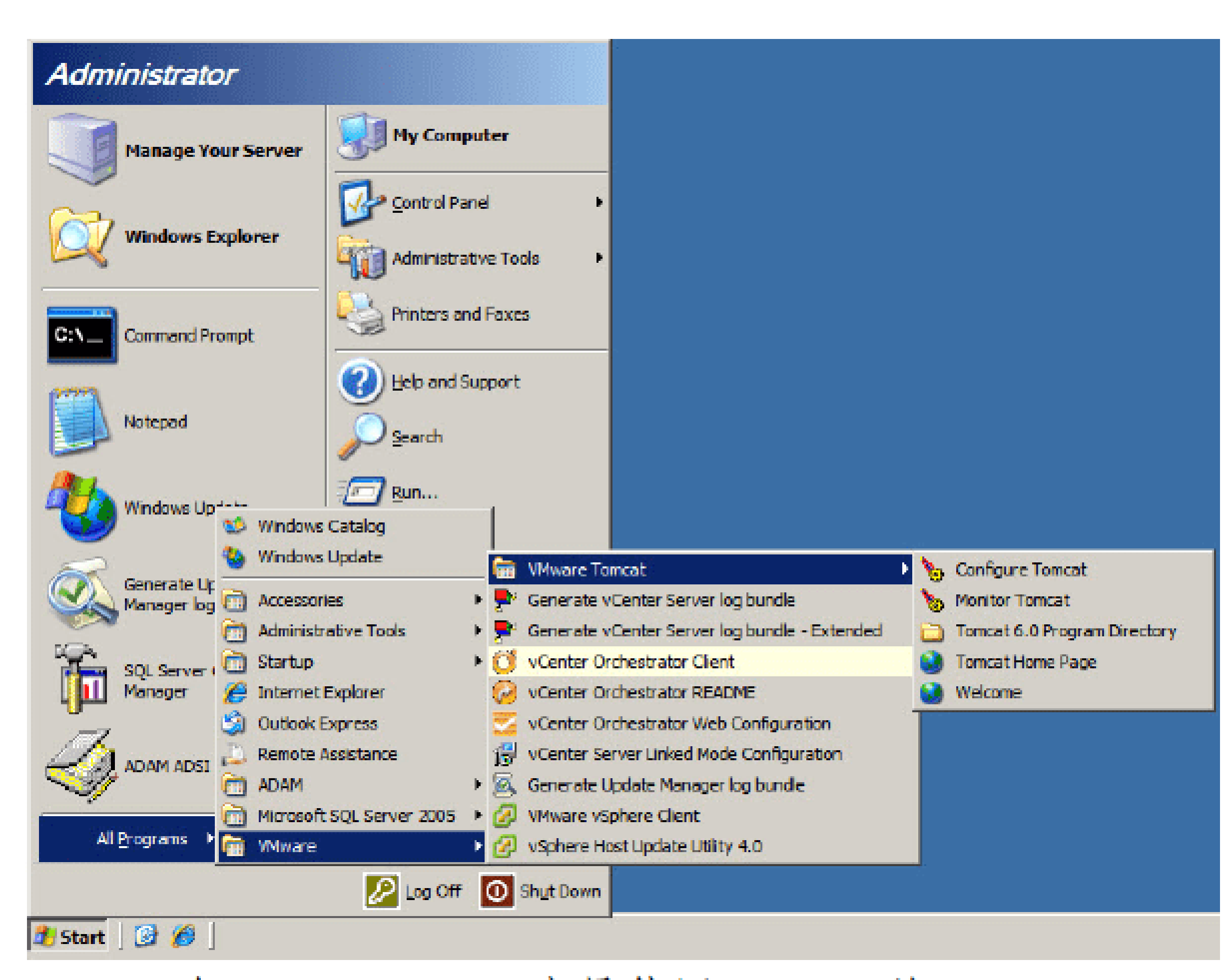
2. 从 vSphere Client 连入 VMware vCenter

在安装 VMware vCenter 时，系统如果发现计算机没有安装 vSphere Client，也会自动将程序安装起来。有了 vSphere Client 之后，我们就可以从 vSphere Client 联机到 VMware vCenter 上，并运行虚拟机的操作。

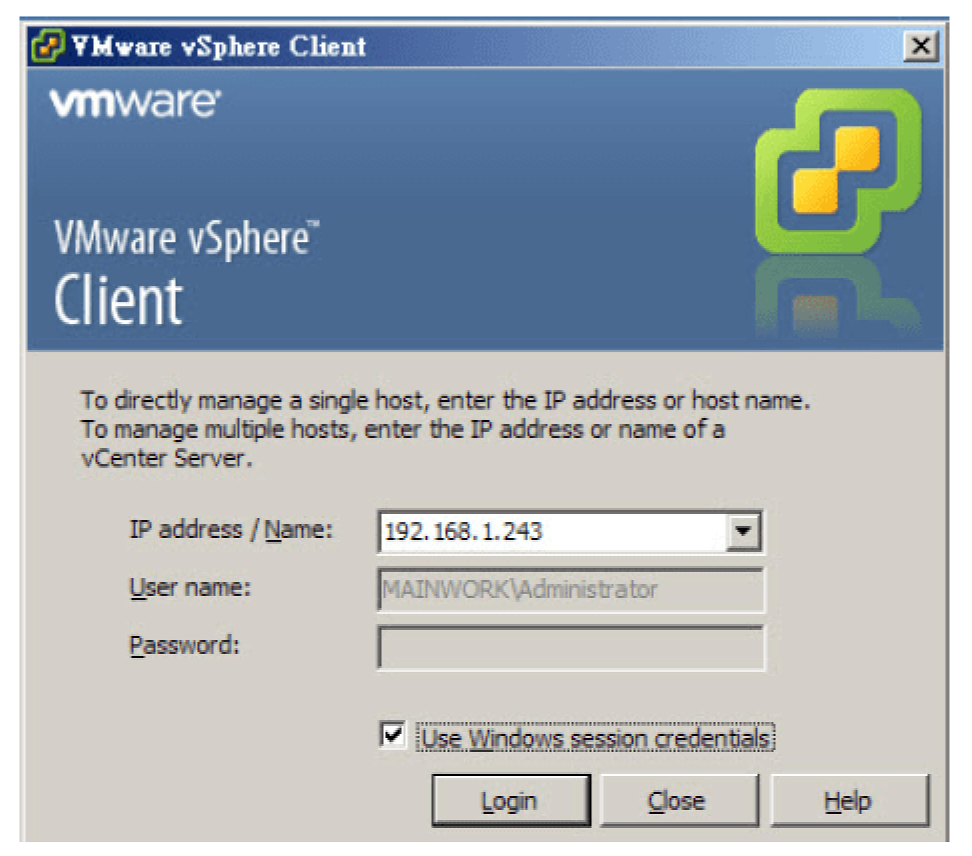


▲ 在 vCenter 的光盘中也有安装 vSphere Client 的选项

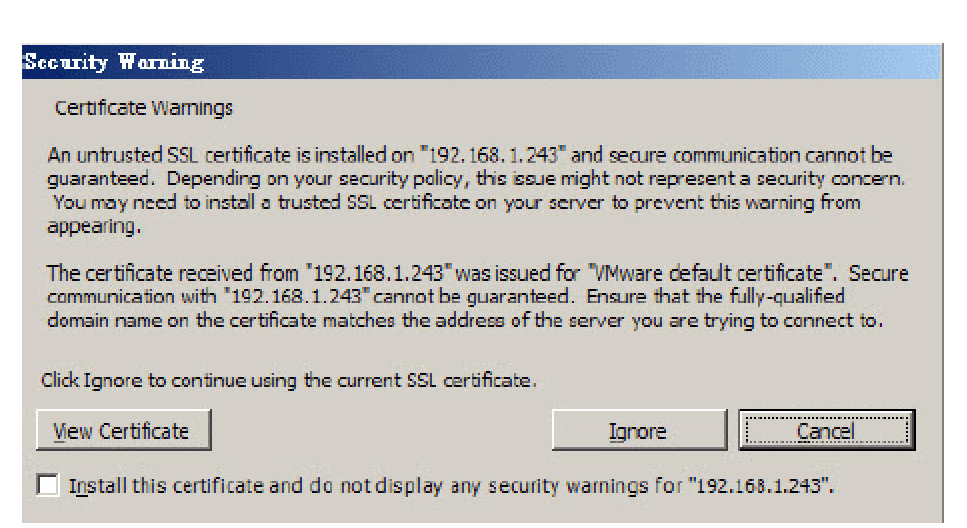
我们在前几章使用了 vSphere Client 连入 ESX/ESXi，那么要总控多台 ESX/ESXi 的话，在 VMware vCenter 下是怎么操作呢？事实上在安装 vCenter 时，系统会在 Windows 2003 下安装一个 Web Server，使用的是 Tomcat，因此要操作 VMware vCenter，也是要利用一个客户端连入，而这个客户端其实就是我们之前已经用得很多的 vSphere Client 了。只要打开 vSphere Client，并且在 IP 的地方键入 VMware vCenter 的 IP，键入在安装 vCenter 时配置的账号口令，就可以直接连入了。



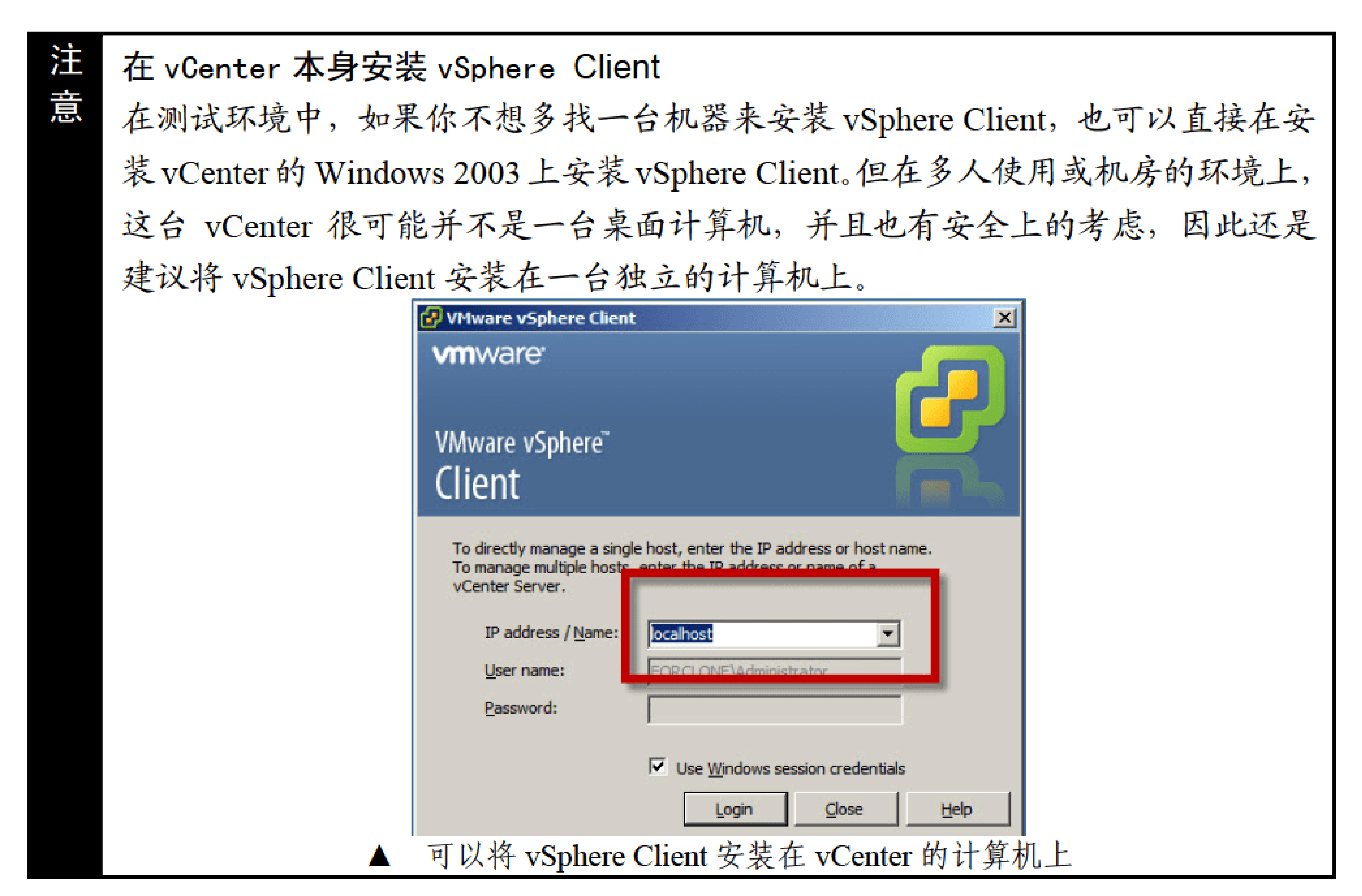
▲ 在 vCenter Server 中是使用 Tomcat 的 Web Server



▲ 还是一样打开 vSphere Client，但确定要键入 vCenter 那台 Windows 2003 的 IP



▲ 进入 vCenter Server 了


注意

在 vCenter 本身安装 vSphere Client

在测试环境中，如果你不想多找一台机器来安装 vSphere Client，也可以直接在安装 vCenter 的 Windows 2003 上安装 vSphere Client。但在多人使用或机房的环境上，这台 vCenter 很可能并不是一台桌面计算机，并且也有安全上的考虑，因此还是建议将 vSphere Client 安装在一台独立的计算机上。

▲ 可以将 vSphere Client 安装在 vCenter 的计算机上

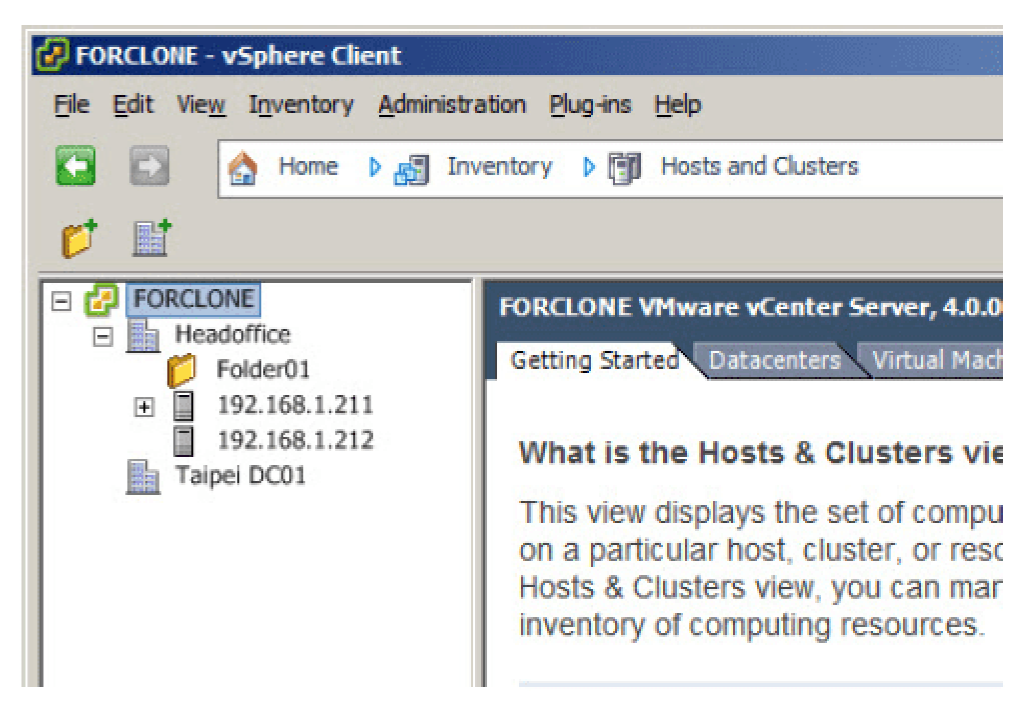
13.4 理解 VMware vCenter 的基本架构

VMware vCenter 是一个多物理 ESX 的管理界面，提供类似 vSphere Client 一致的操作方式，可以进行多台 ESX 物理服务器的管理，当然也可以从其上运行 VMotion 等高级功能，但是在开始之前，我们必须理解 VMware vCenter 的基本使用方式，我们就从登录 VMware vCenter 开始讲解。

13.4.1 了解 vCenter 的部件配置

VMware vCenter 也是使用层次式的树形架构，以方便集中管理不同的 ESX/ESXi 主机。当我

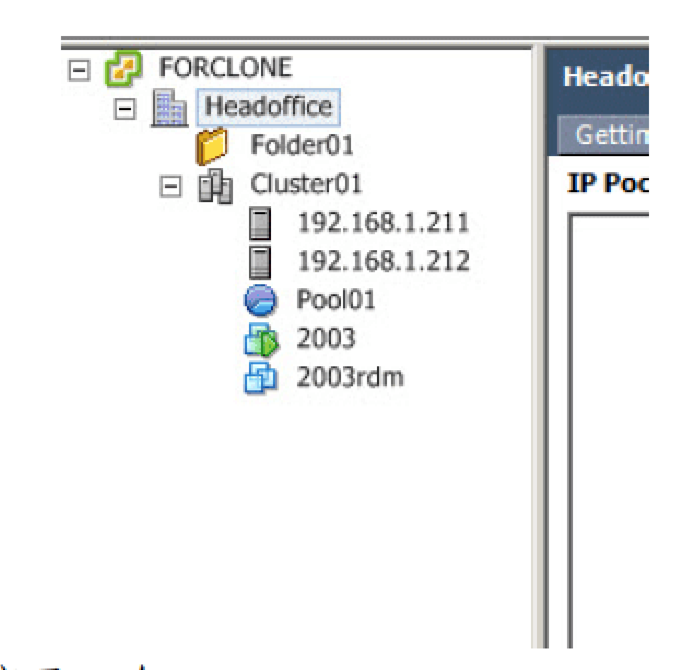
们进入 VMware vCenter 之后，可以看到左边就是部件管理的界面。在 vCenter 中，管理界面的组织分成下面几个单位：



▲ 这就是部件视图画面

▶ vCenter 中的管理单位

- Datacenter（数据中心）：Datacenter 一般来说是以机房位置来划分。Datacenter 是 vCenter 中最高级的划分单位。
- Cluster（集群）：多台 ESX/ESXi 计算机可以组成一个集群以方便进行多种高级功能。一般来说，我们会将同一机房的计算机放入同一个 Datacenter，但是会将同一网段的计算机放置到一个集群中以方便管理。我们会在稍后的章节对 Cluster 有更进一步的说明。
- Host（主机）：这里指的是 ESX/ESXi 主机。你可以将 Host 加入一个 Datacenter，也可以加入一个 Cluster。
- VM（虚拟机）：就是一个 VM。VM 可以放在 Host 中，也可以放在 Cluster 中，只要你的 Cluster 有配置 HA 或 DRS 等高级功能，我们会在后面的章节进行说明。
- Folder（文件夹）：Folder 是一个方便管理的抽象单位，你可以创建 Folder 来存放同一层级的部件，如你可以创建一个 Folder 来存放好几个 Datacenter，或是在 Datacenter 之下，创建一个 Folder 来存放好几个 Cluster 等，以此类推。

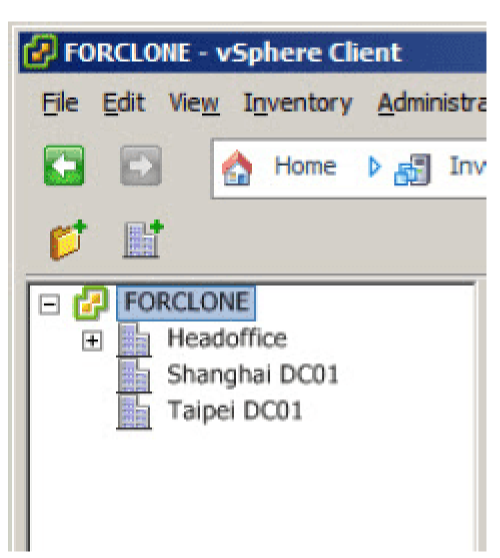


▲ 较完整的演示，有 Datacenter，Folder，Cluster，Host 及 VM

13.4.2 VMware vCenter 的基本单位：Datacenter

VMware vCenter 的界面和 vSphere Client 的界面几乎完全一样，但 VMware vCenter 是多台 ESX 的组合，因此其数据的呈现方式可以说是 ESX 界面的超集（Superset）。举例来说，在 vSphere Client 的界面中，我们在窗口左边看到的是虚拟机或是资源池（用来规定某些虚拟机只能使用固化的资源）。但是在 VMware vCenter 的界面中，管理的界面不但上升到多台物理机，更可以上升

到机房、业务单位、地点、国家等。这样的好处是让你方便管理，并且可以依照具体的机器分布情况来管理。



▲ 可以根据具体的位置来配置 Datacenter

在 VMware vCenter 中，所有的存储单位都称为 Datacenter。一个 Datacenter 可以放置多个 ESX 物理服务器或是 Cluster，而每一个 ESX 物理服务器上也会有自身的虚拟机。因此当第一次进入 VMware vCenter 时，系统会要求你创建第一个 Datacenter，以便能存放 ESX 物理服务器和其上的虚拟机。Datacenter 允许树形结构，意思就是你可以在一个 Datacenter 中创建子 Datacenter，并且依照其逻辑位置来存放物理的 ESX 服务器，这一点会让整个 vSphere 架构更贴近物理业务的架构，十分方便系统管理员的操作。

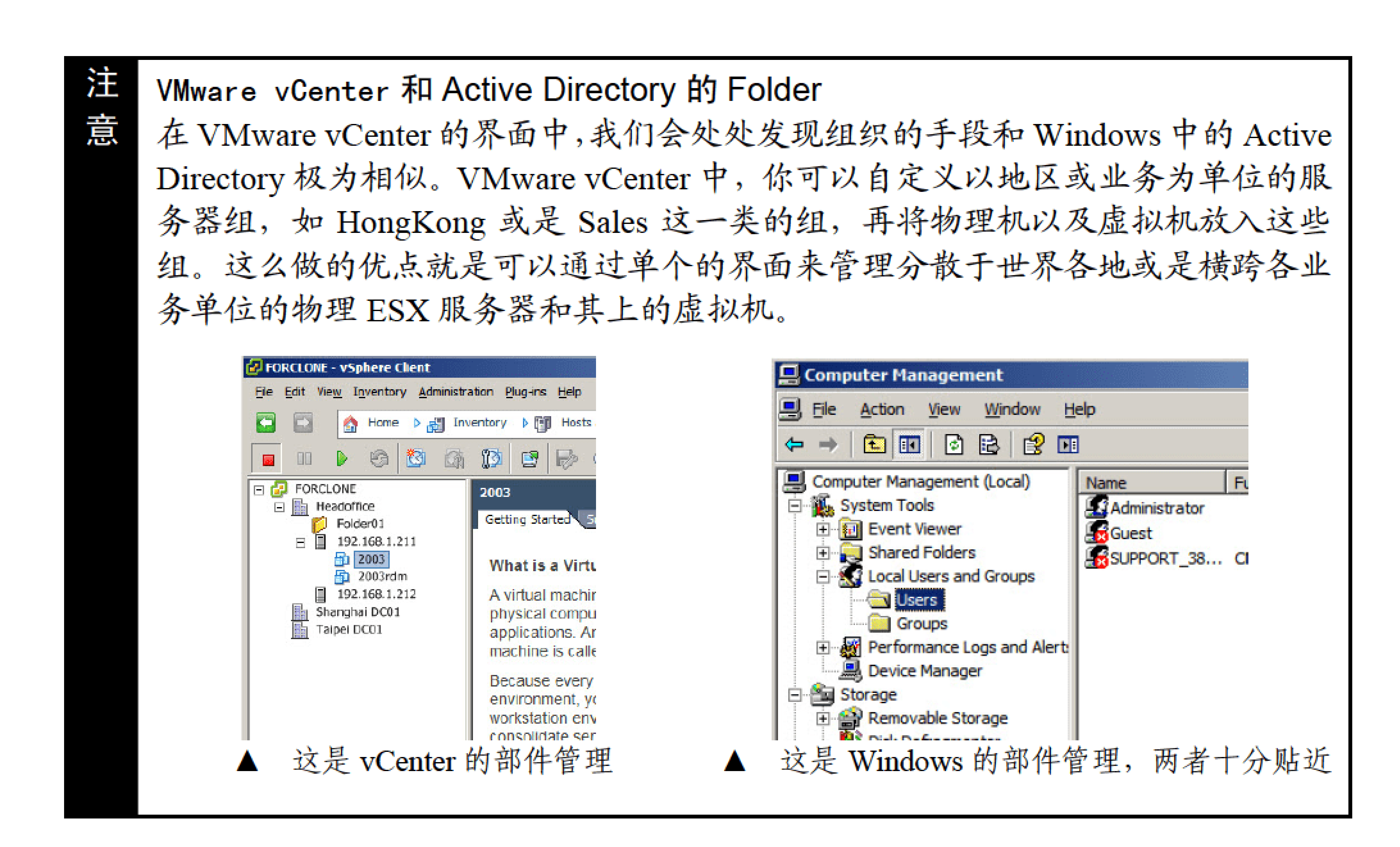


▲ 第一次进入 vCenter 时，系统会要求创建一个新的 Datacenter

注意

VMware vCenter 和 Active Directory 的 Folder

在 VMware vCenter 的界面中，我们会处处发现组织的手段和 Windows 中的 Active Directory 极为相似。VMware vCenter 中，你可以自定义以地区或业务为单位的服务器组，如 HongKong 或是 Sales 这一类的组，再将物理机以及虚拟机放入这些组。这么做的优点就是可以通过单个的界面来管理分散于世界各地或是横跨各业务单位的物理 ESX 服务器和其上的虚拟机。

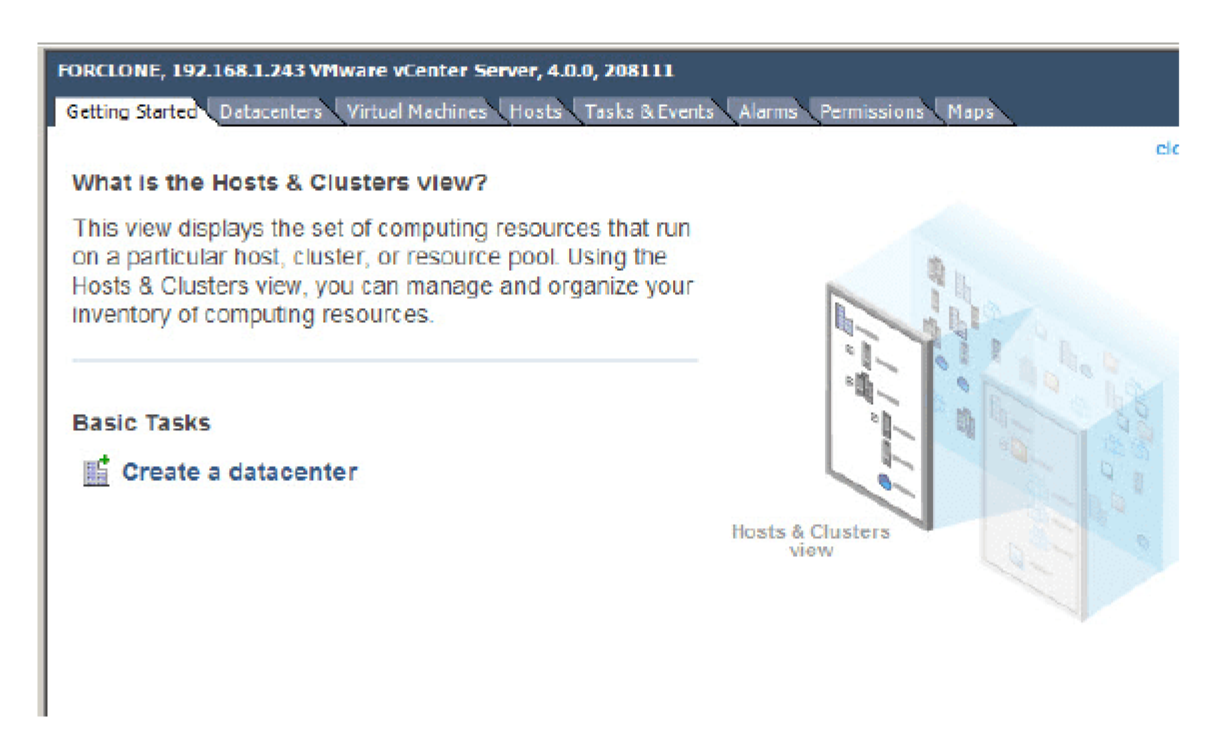


▲ 这是 vCenter 的部件管理　　▲ 这是 Windows 的部件管理，两者十分贴近

▶ 创建第一个 Datacenter 及主机

在进入 VMware vCenter 的界面之后，首先映入眼帘的就是 Getting Started。这里是所有任务的第一步，就是创建第一个存放服务器的 Datacenter 开始，下面就来看看如何创建 Datacenter。

1. 进入 VMware vCenter 的画面，当前还没有创建 Datacenter，因此在左边的窗格中也没有任何数据。选择 Getting Started/Create a datacenter 选项。



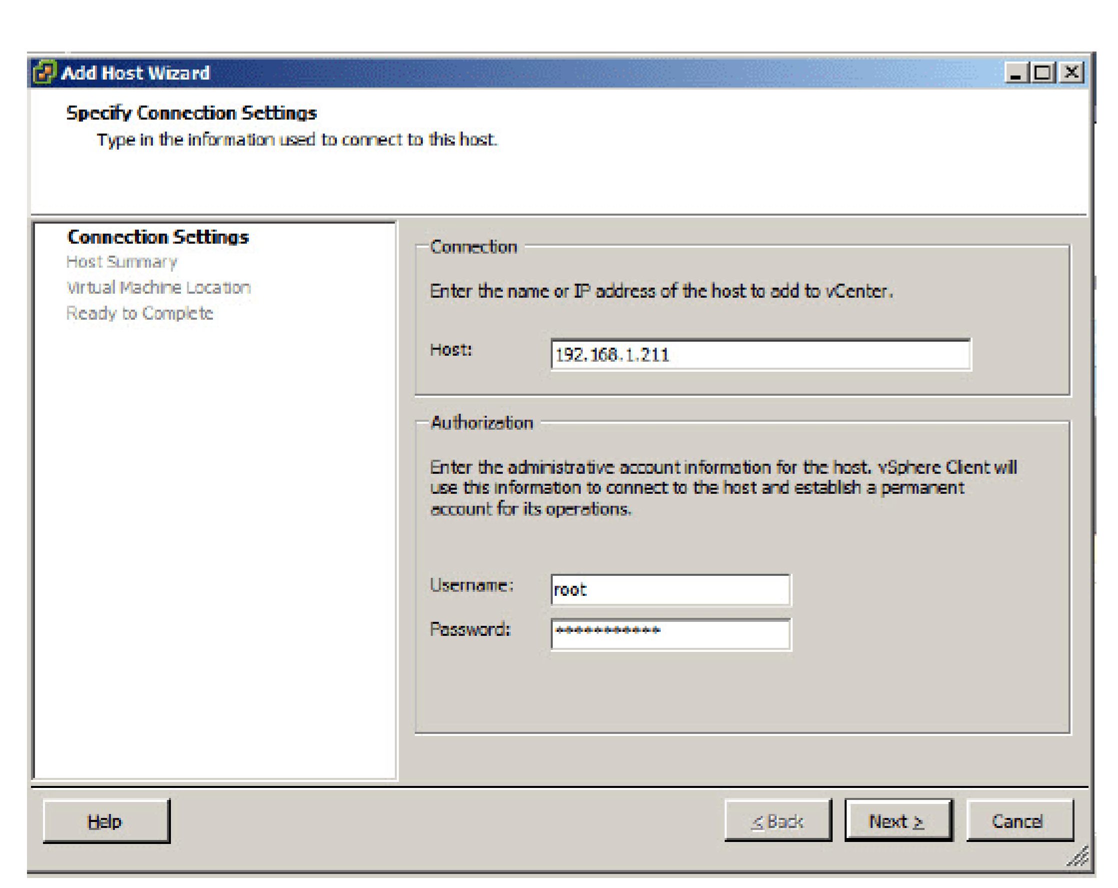
▲ 选择创建新的 Datacenter

2. 此时系统会在左边创建一个 Datacenter，你必须键入 Datacenter 的名字，如 Headquarters。
3. 接下来的动作就是将 ESX 物理服务器加入这个新建的 Datacenter。选择 Add a host 选项。



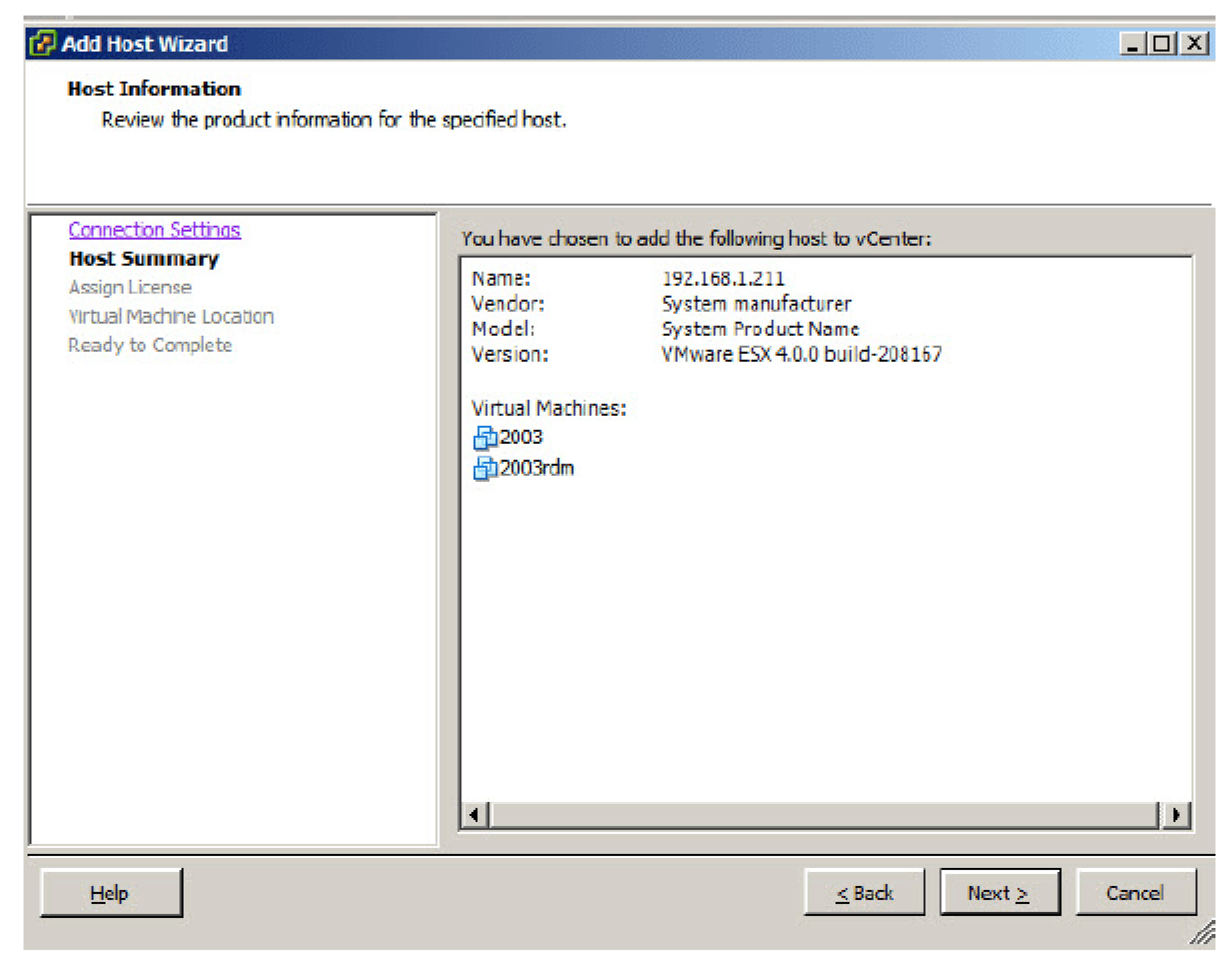
▲ 键入 Datacenter 的名字　　▲ 新建主机

4. 此时会弹出加入 ESX 服务器的窗口，你可以键入 IP 或是主机名。由于我们在 DNS 中都配置完毕了，VMware vCenter 会读取所在的 Windows Server 2003 的 DNS。另外也要键入加入的 ESX 服务器的账号及口令。
5. 加入成功后会弹出如下图所示的窗口，会给出该服务器的基本数据。单击 Next 按钮继续。



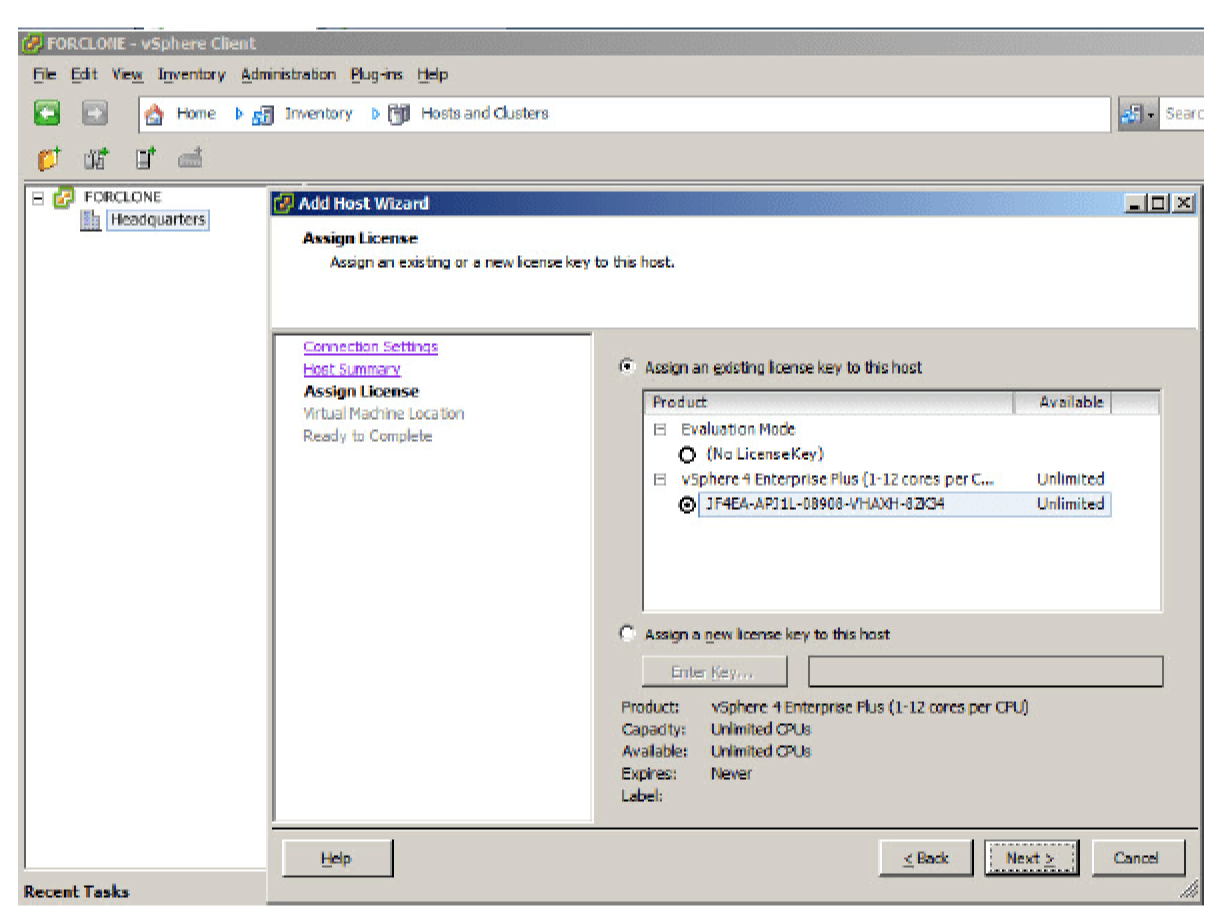
▲ 键入主机基本数据

6. 之后系统会询问这台 ESX 物理服务器要置放在哪一个 Datacenter 中。当前我们只有一个 Datacenter，就选择置放在刚才创建的 Headquarters 中即可。然后单击 Next 按钮。



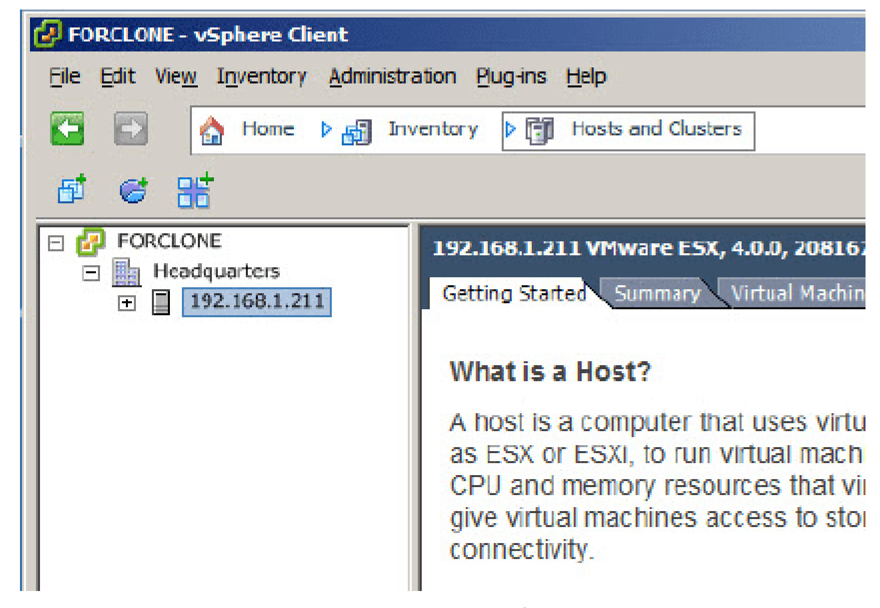
▲ 给出基本数据

7. 之后会再给出一次 ESX 物理服务器的数据，单击 Next 按钮继续。



▲ 再给出一次 ESX 主机授权数据

8. 之后系统会开始做一些加入的动作，接着就可以看到第一台 ESXi 已经加入这台 vCenter 了。



▲ 加入主机了

13.4.3　理解 vCenter 布局

当进入 vCenter 之后，我们可以看到熟悉的画面和 vSphere Client 完全一样。这就是整个 vSphere 的总控面板，所有的动作及功能都是在此落实。在 vCenter 中，每一个布局都有其功能，我们就来看看每个部分的功能。

▶ vCenter 中不同的局块

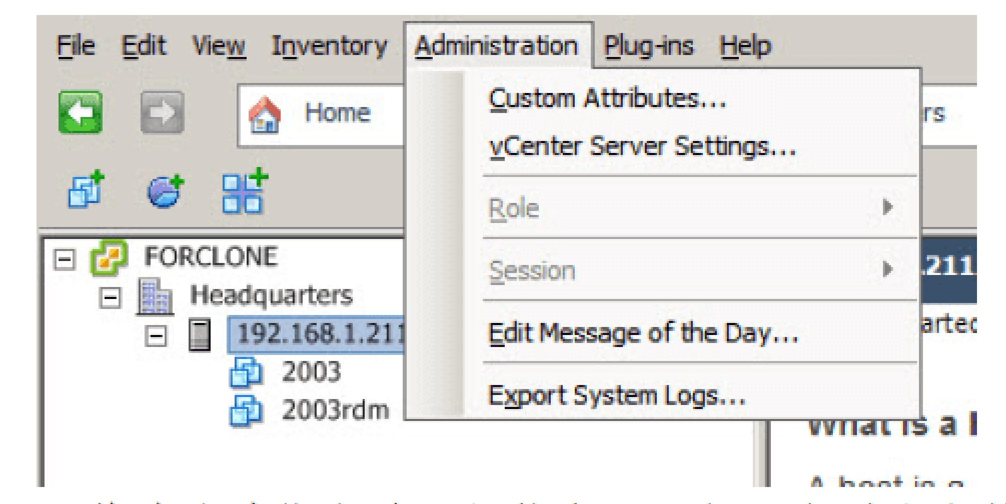
- 菜单
- 功能导航栏
- 功能仓库搜索（Inventory Search）
- 工具条（会随不同部件而改变）
- 部件窗口
- 功能窗口
- 当前任务指示栏



▲　完整的 vCenter 布局

1. 菜单

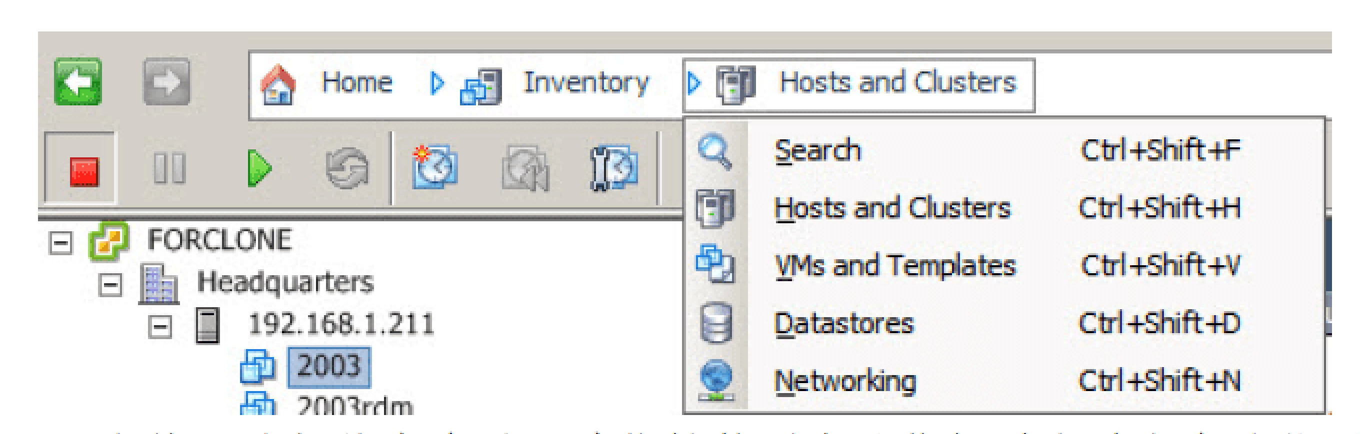
所有 vCenter 下的事情在菜单中都可以落实，但大部分菜单的功能都已经在整个 vCenter 的窗口中都有相应的工具条或是快捷键来表示。



▲ 菜单的功能当前已经较少用，但一定对比完整

2. 功能导航栏

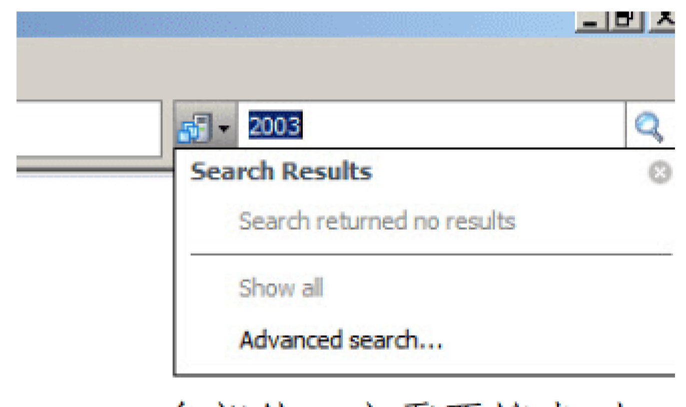
这里会给出当前你所在的功能部分，以方便操作。vCenter 的界面和我们平常使用的 Windows 浏览器很像，也可以上一步或下一步，这也是为什么 vCenter Server 主要是安装在 Windows 的原因之一。



▲ 当管理的部件变多时，功能导航列会让你知道当前自身的位置

3. 功能仓库搜索（Inventory Search）

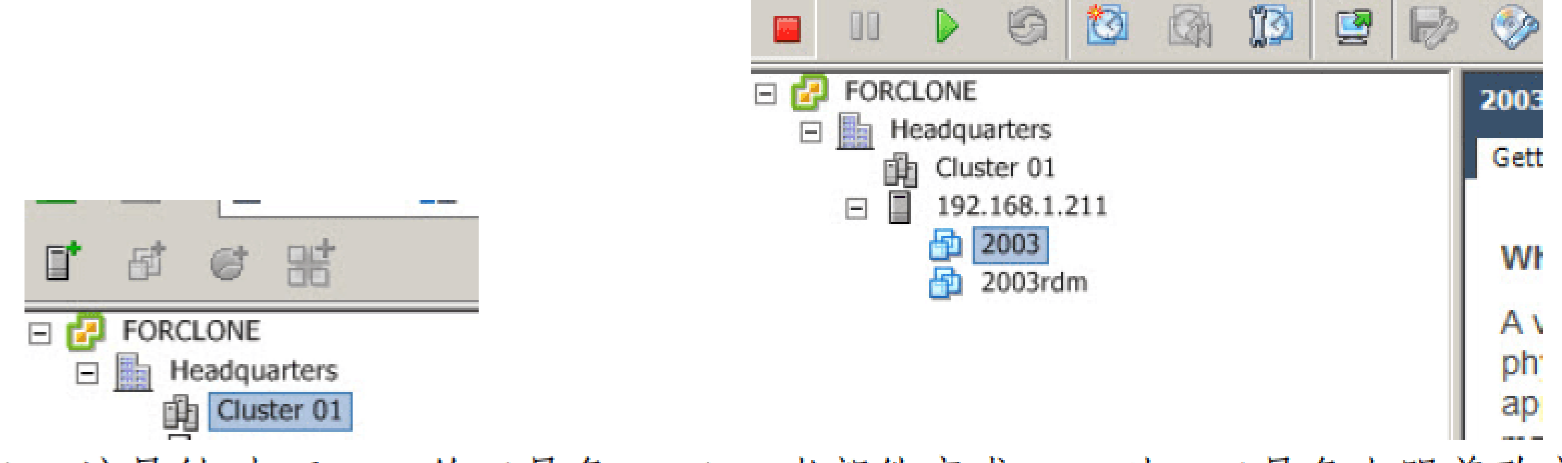
当你管理的 Datacenter 有好几个，管理的 Cluster、主机、VM 越来越多时，层次也越来越深，就会需要这个搜索功能来找到特定的 ESX/ESXi 或 VM 等。



▲ 多部件一定需要搜索列

4. 工具条

工具条是平常使用最多的地方，大部分的功能都在工具条上落实，当你在整个 vCenter 中点选不同部件时，工具条也会随之改变。

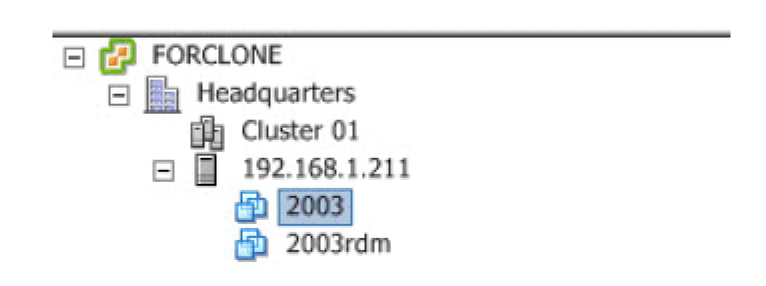


▲ 这是针对 Cluster 的工具条　▲ 当部件变成 VM 时，工具条也跟着改变

5. 部件窗口

这里是整个 vCenter 最重要的部件浏览窗口，所有的 Datacenter、Cluster、Host、VM 等都在这里显示。在 vSphere 中，你也可以利用拖拉的方式对这里的主机、Cluster 或是 VM 进行操作，

如 VMotion 就可以直接移动 VM，在稍后的章节会有完整的介绍。



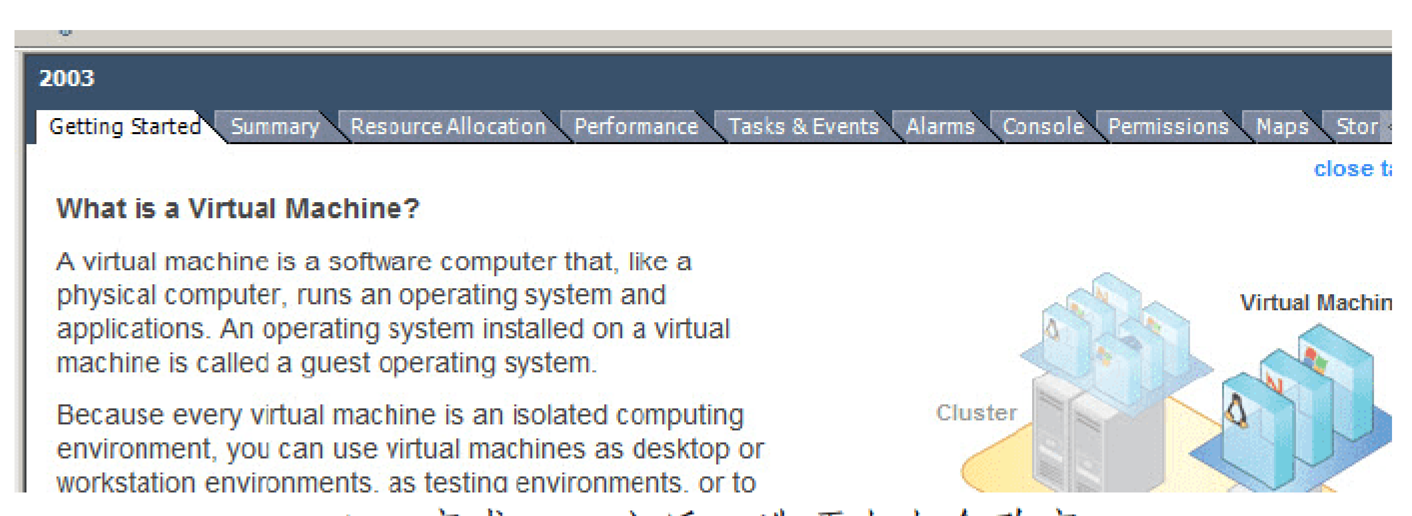
▲ 针对 Container 类对象的操作都在这个窗口

6. 功能窗口

当在部件窗口选择不同的部件时，这里的功能窗口就会随着改变。功能窗口是所有功能及配置的发生所在，并且会有完整的信息，在下一节会针对功能窗口有更完整的说明。



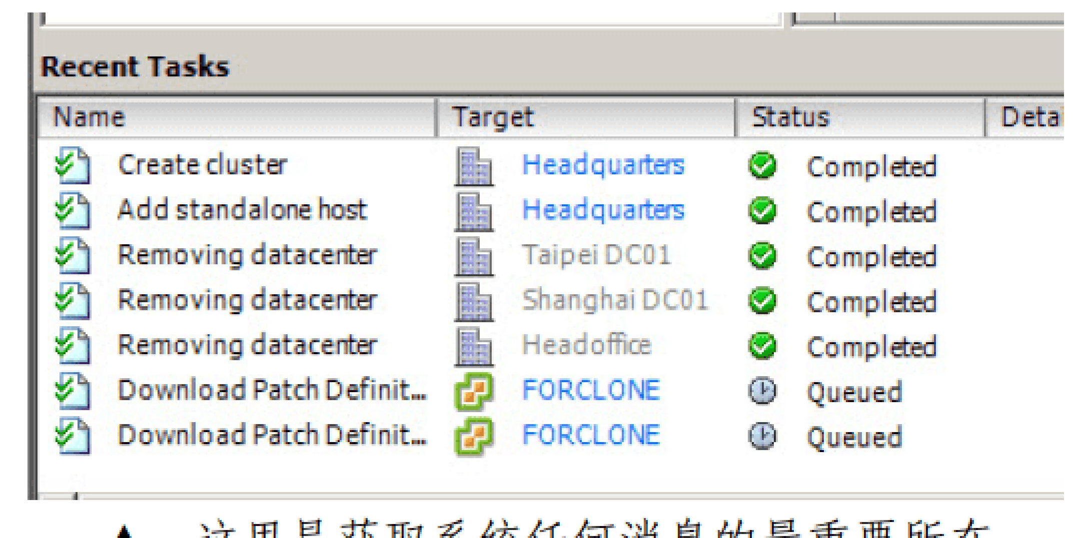
▲ 这里是每一个部件的功能使用



▲ 变成 VM 之后，选项卡也会改变

7. 当前任务指示栏

当你进行了任何动作，在这里都会有进度和落实的指示，特别要注意的是，如果有两个以上的用户登录到 vCenter，所有用户所做的动作都会在这里显示，因此常常会有一些其他消息。



▲ 这里是获取系统任何消息的最重要所在

Tasks Alarms

▲ 也可以直接检视报警消息

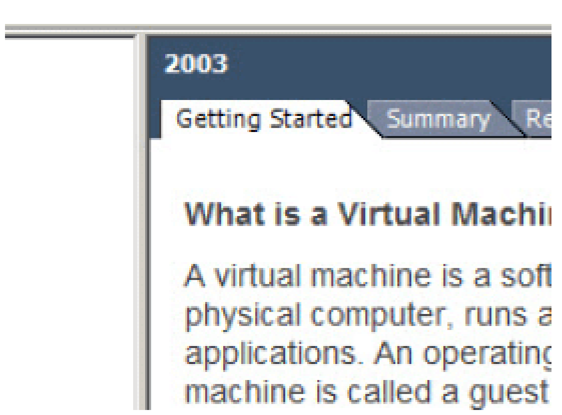
13.4.4 理解 vCenter 的部件：vCenter 的选项卡

在前一小节，我们已经进入 vCenter 并创建了一个 Datacenter 及主机。事实上，在 vSphere vCenter 中最重要的功能性部件是 Cluster，我们会在稍后的章节说明。但 Datacenter 的重要性是存

放不同的 Cluster 及主机，因此能理解 Datacenter 的各个部件才能完整使用，下面就来看看每一个部件的功能，首先从 vCenter 主机开始。

1. Getting Started 选项卡

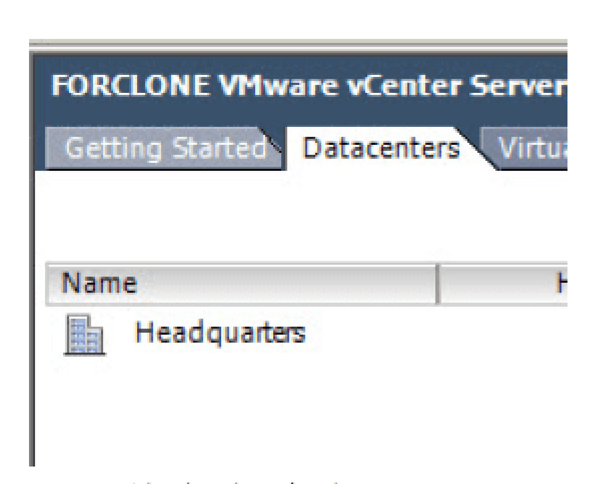
当我们一进入 vCenter 时，会先进入这个选项卡。这个选项卡在每一个 VMware 的部件中都有，会依据不同的部件（如 vCenter、Datacenter、Cluster、Host、VM 等）有不同的内容，通常这里的功能都是在进入该部件时的第一件要做的事。在 vCenter 中，当前第一件事要做的就是创建 Datacenter，我们在上一小节已演示。



▲ 每一个部件都有这个 Getting Started 选项卡

2. Datacenter 选项卡

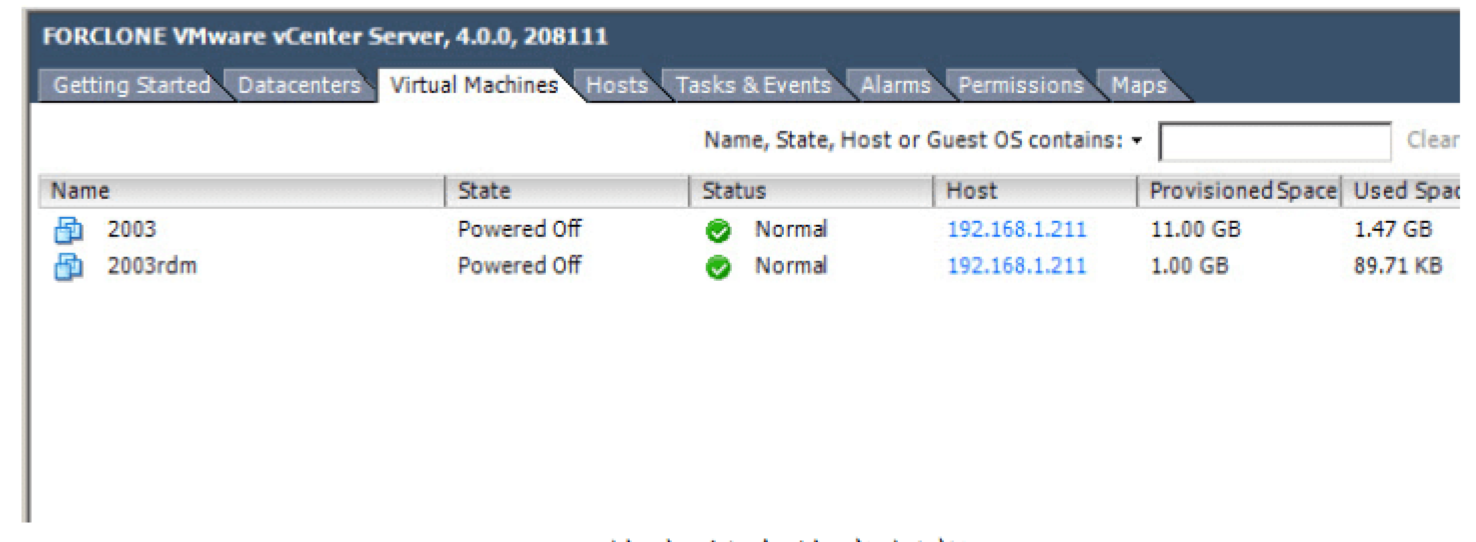
这里会给出当前这台 vCenter 所管理的 Datacenter。当你在给出的 Datacenter 上右击时，也可以直接针对这台 Datacenter 进行操作。



▲ 给出当前的 Datacenter

3. Virtual Machines

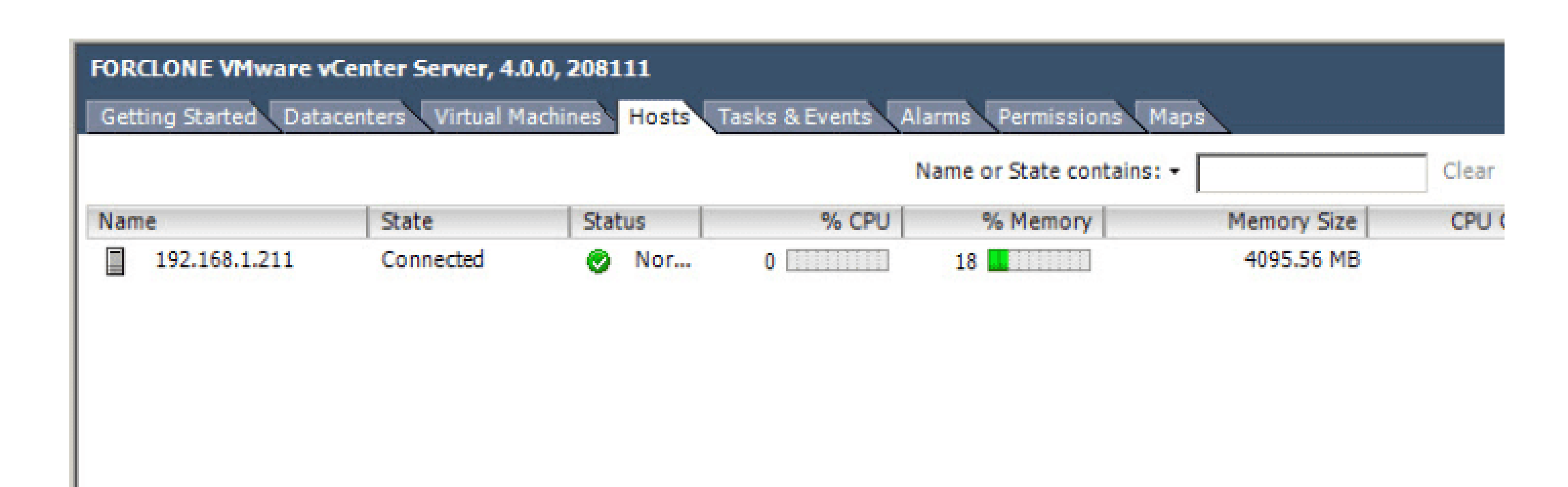
这个 Datacenter 下所有主机上的 VM 列表，并且会有完整的信息，如所属的主机以及磁盘占用空间、使用的 CPU/内存资源等。



▲ 给出所有的虚拟机

4. Host

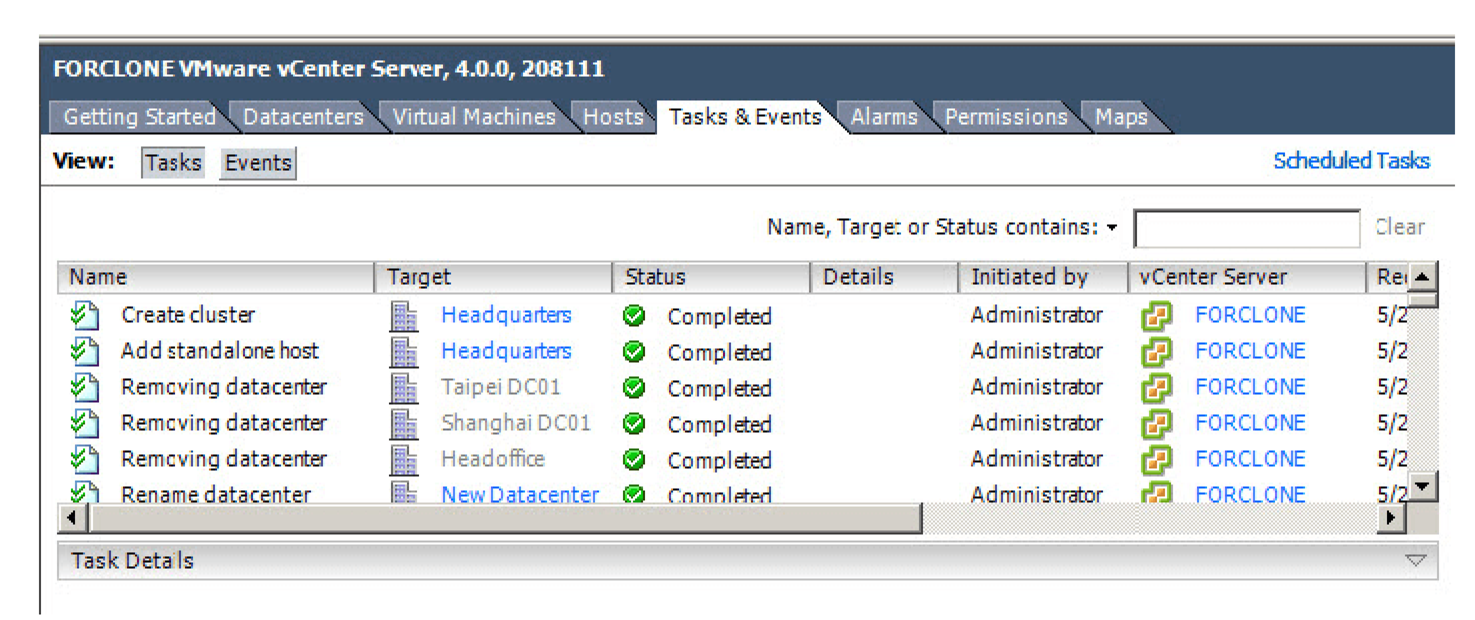
这个 Datacenter 下所有的主机列表，这里也会给出当前使用的资源，并且也可以利用右键对单个主机进行操作。



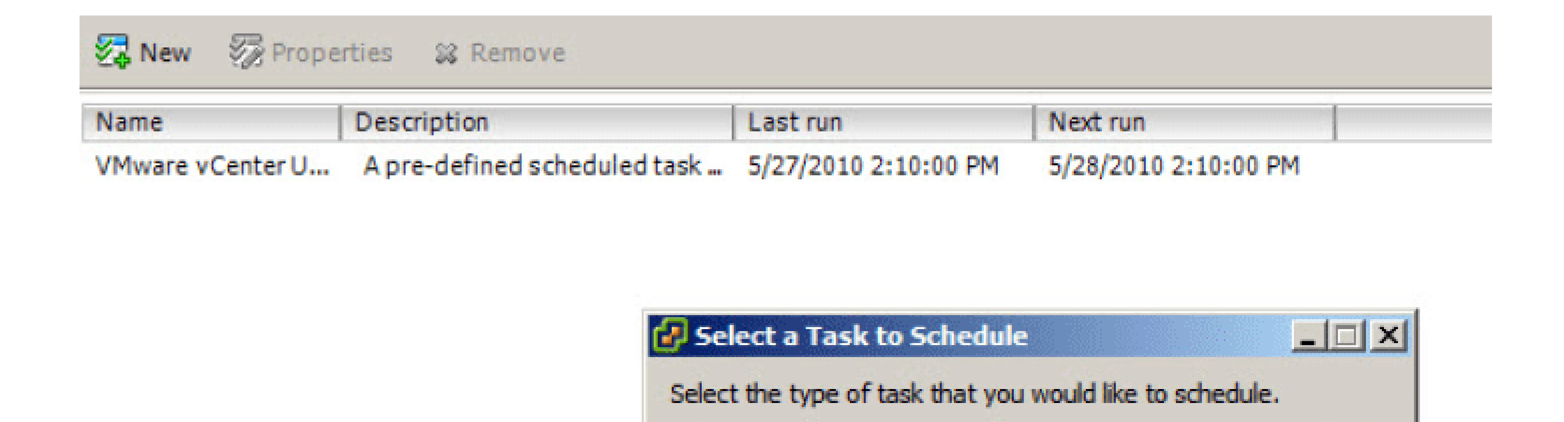
▲ 给出这个 vCenter 下所管理的所有的 ESX/ESXi 主机

5. Task & Events

这里会表项所有在这台 vCenter 中发生的所有任务及警示消息，你也可以在这里激活排定的任务，落实一些系统自动化的事情。



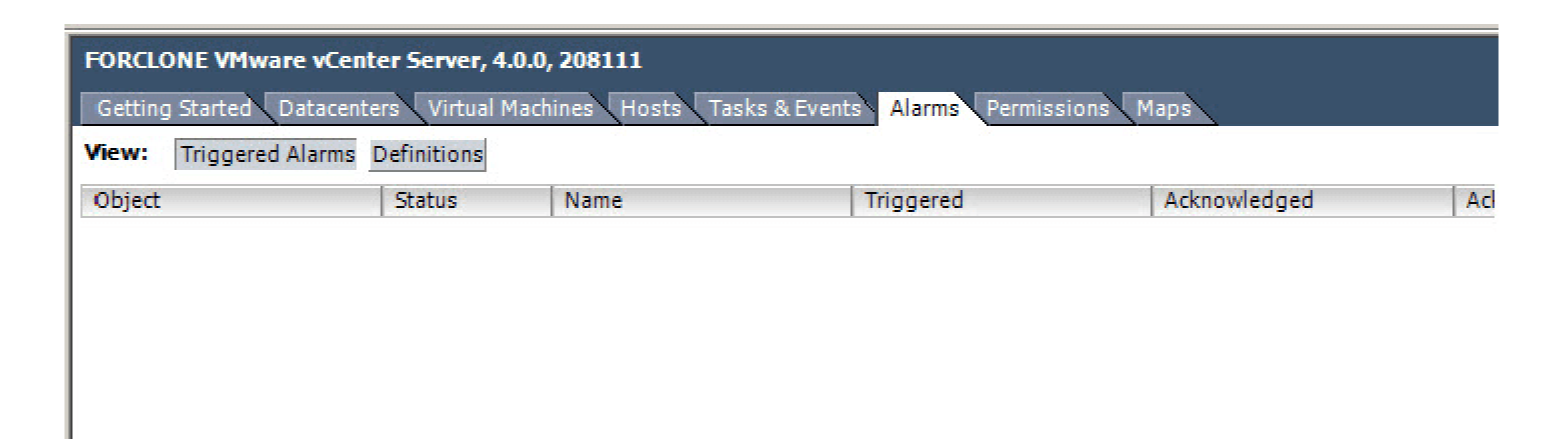
▲ 这里会给出所有做过的任务



▲ 你也可以激活一个新的任务，并且给定时间落实

6. Alarms

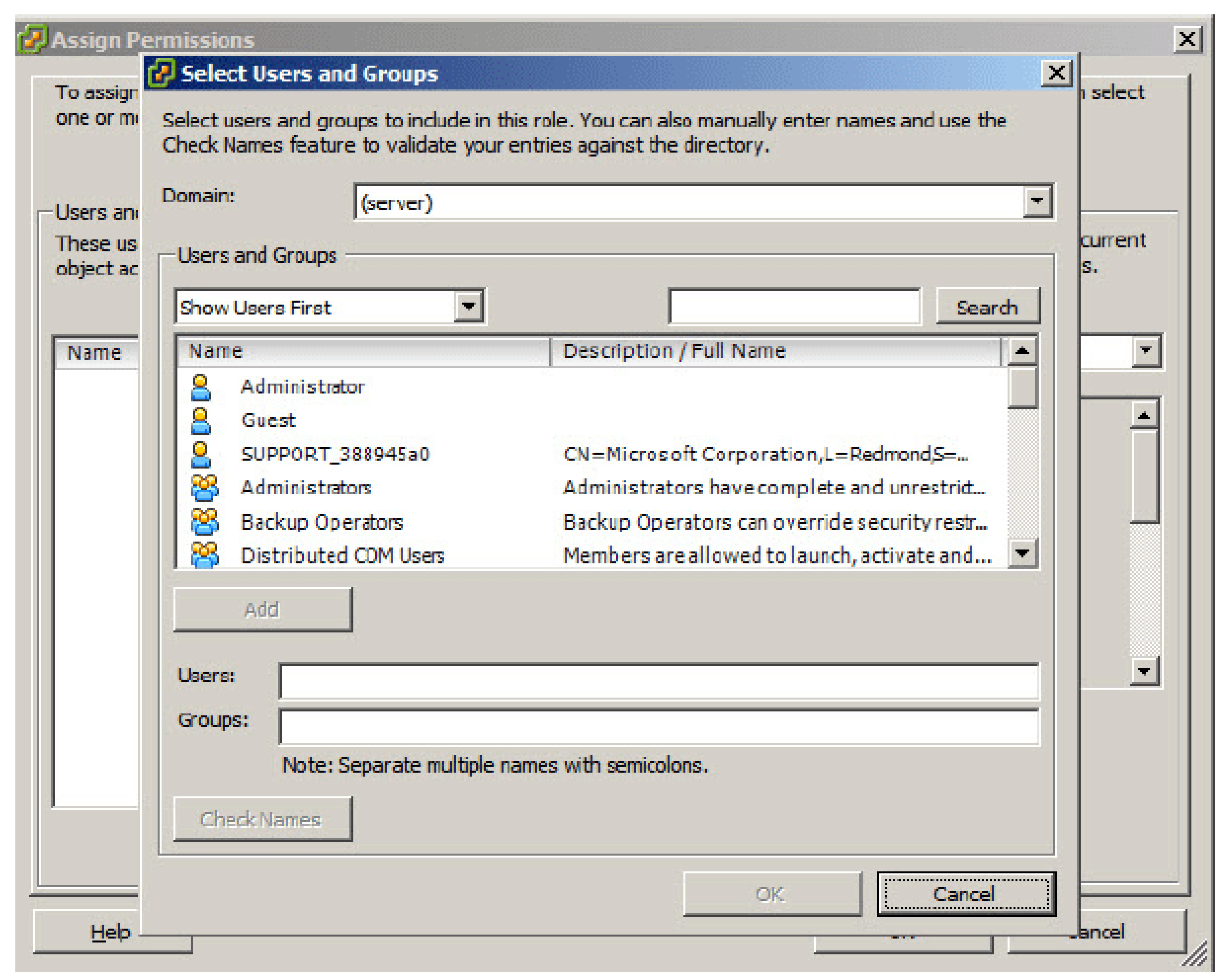
这里会显示在系统中发生过的报警消息，如果你在操作 vCenter 中有任何错误，一定要来这个地方检视完整的消息才能找到解决方案。



▲ 给出系统的报警消息

7. Permissions

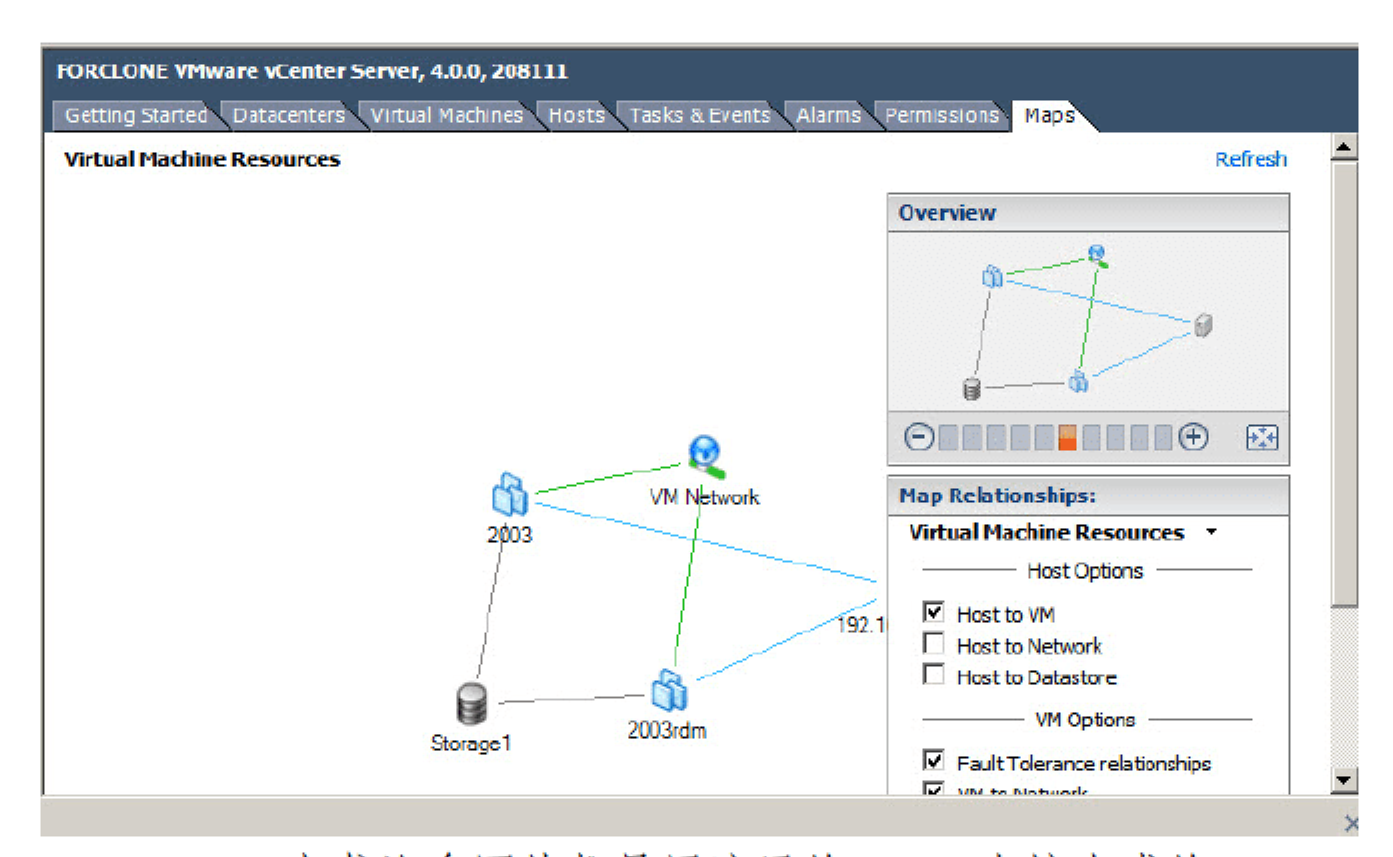
这里是配置 vSphere 权限的地方，vSphere 的权限也和 Windows AD 的观念很贴近，根据不同的部件，不同的用户，规定不同的操作。在一个多人管理的 vSphere 环境中，权限是十分重要的。



▲ 权限是总控 vCenter 的重要基础，多人环境中就必须要配置了

8. Maps

vSphere 中的布局图功能是系统管理员最重要的工具，能将整个环境中的所有部件都用最直观的方式联系起来。在这个部分有许多选项，让你将不同的 vSphere 物理或网络加入 Maps 中，这在解释企业 IT 架构时，能让系统管理者更方便，也提供了更直观的说明。



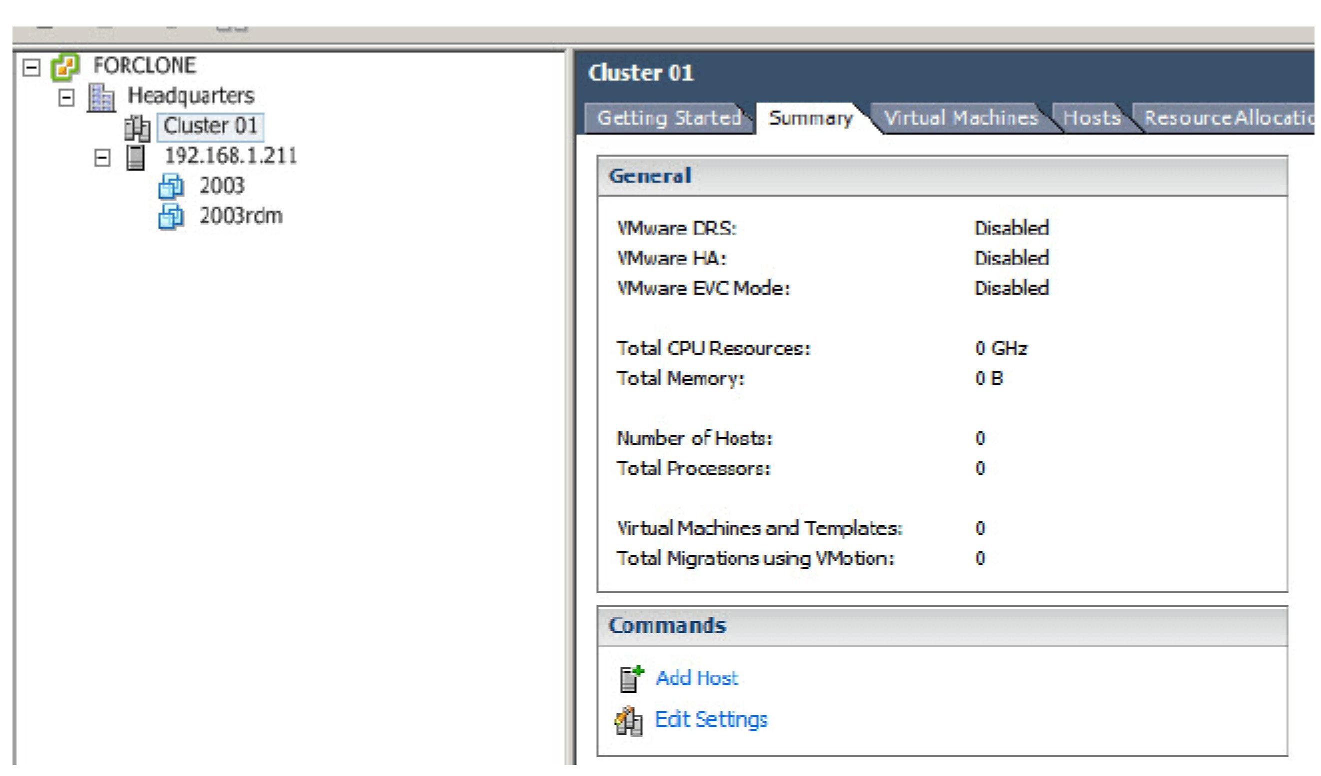
▲ 本书许多图片都是用这里的 Maps 直接生成的

13.4.5 其他部件的选项卡

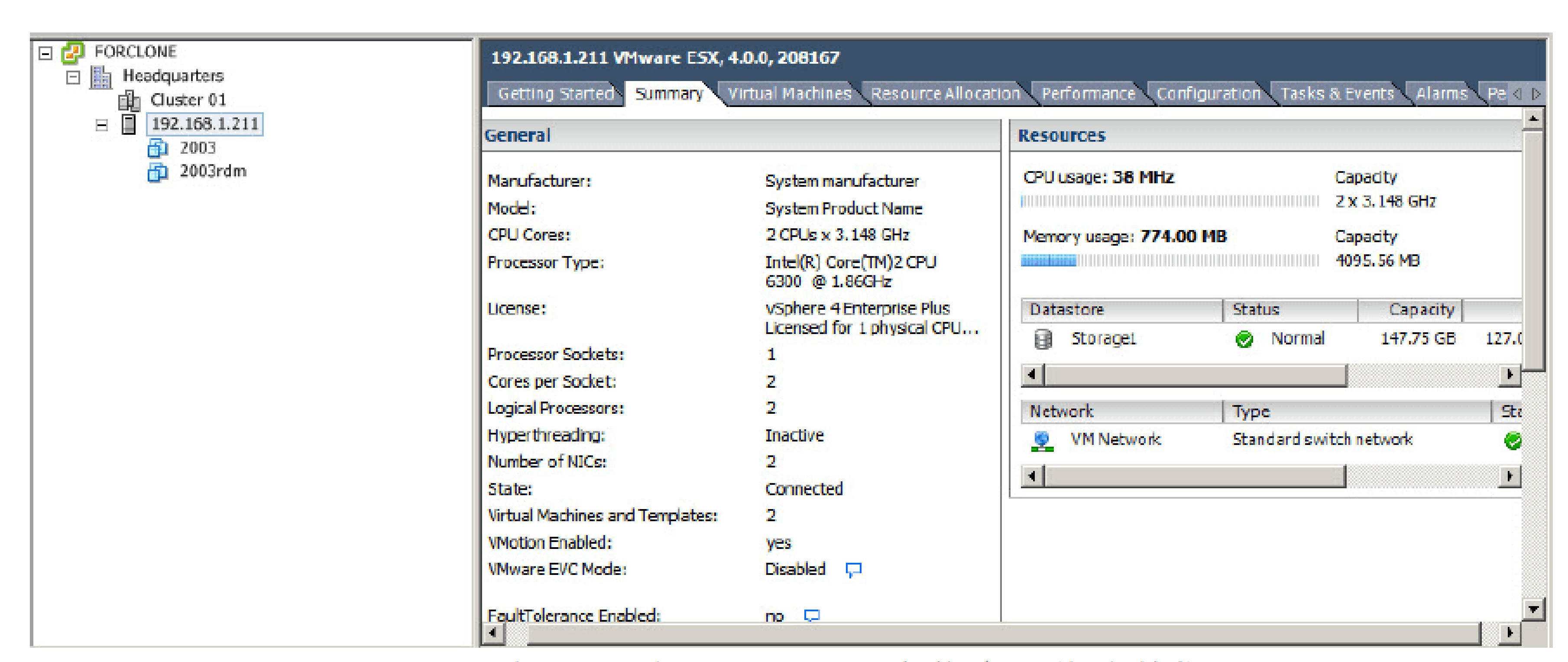
在 vSphere 中，部件窗口中的所有 Datacenter、主机、Cluster 以及 VM 在打开时，都会对应到自身的功能窗口。其中许多功能选项卡和 vCenter 的一样，下面就来看看几个较重要的选项卡。

1. Summary

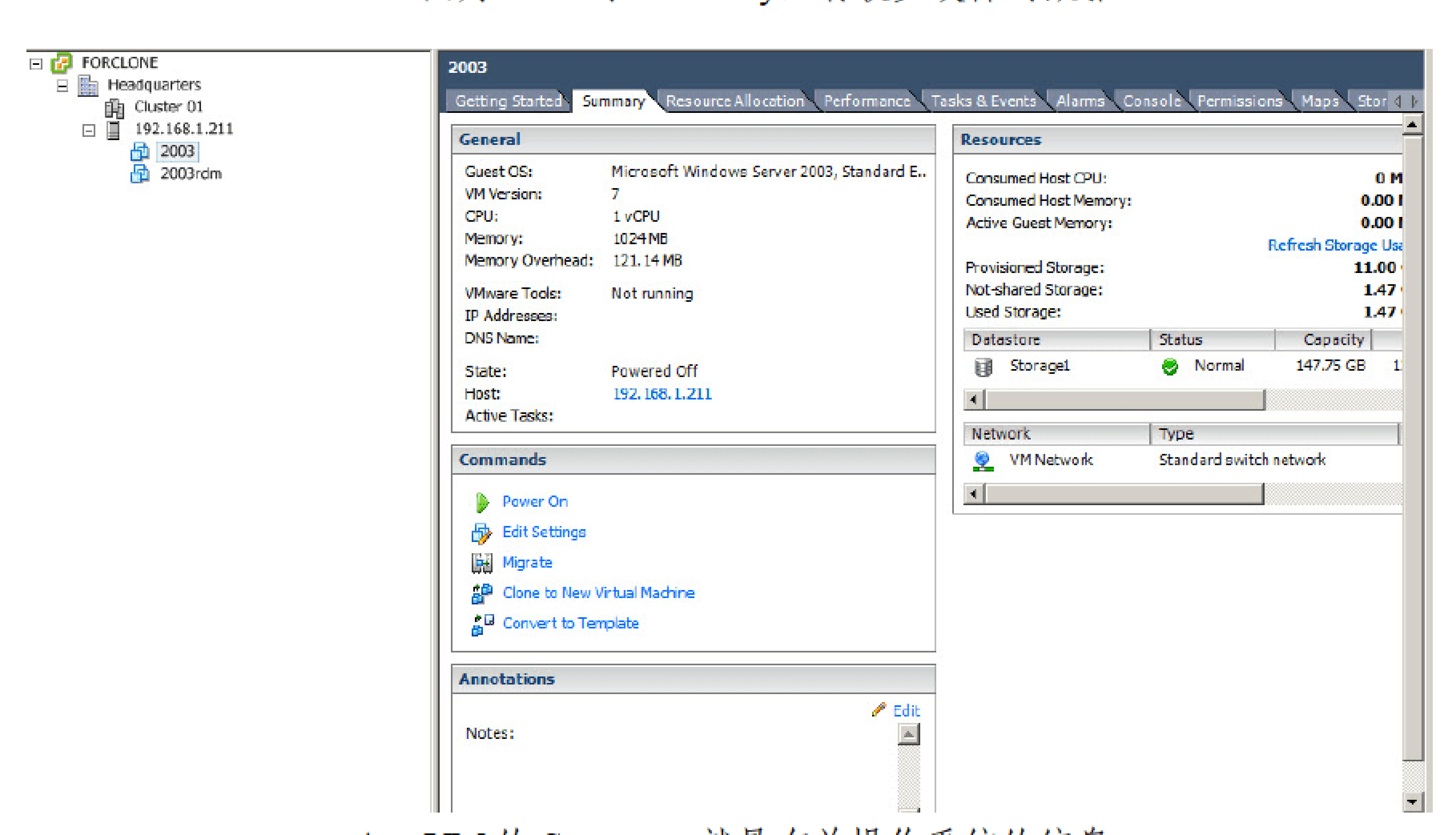
这是在 Datacenter、Cluster、Host 和 VM 部件中都有的功能选项卡。会显示出这个部件的硬件信息和资源使用状况。并且根据部件的不同，会有该部件的操作命令。



▲ 不同的部件也会有不同的 Summary，图为 Cluster 的 Summary



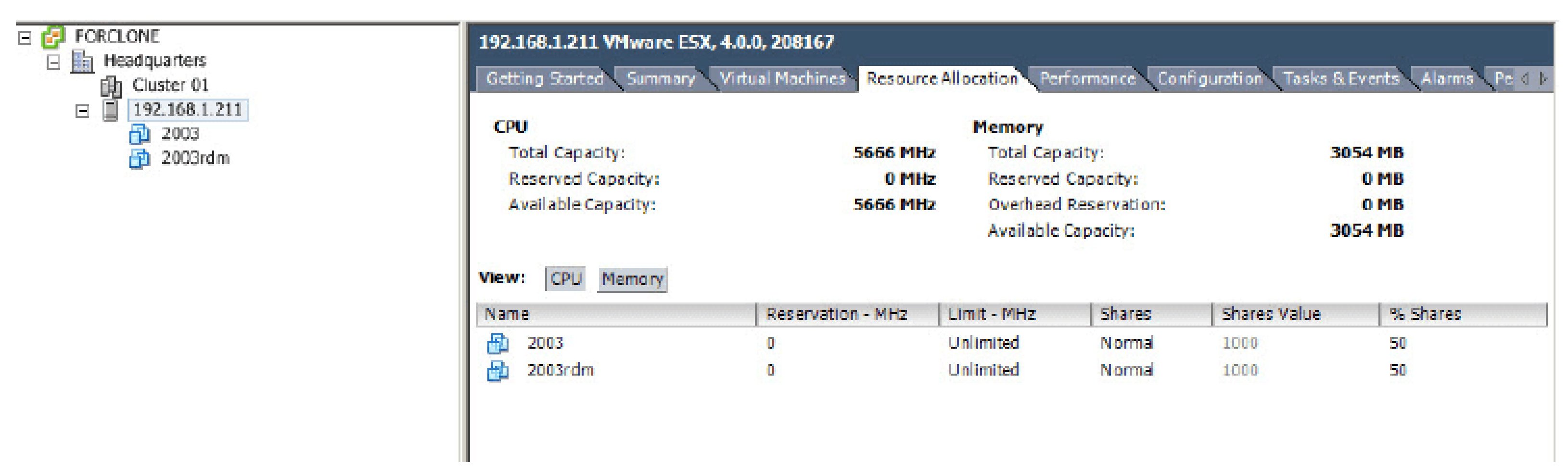
▲ 图为 Host 的 Summary，有较多硬件的数据



▲ VM 的 Summary 就是有关操作系统的信息

2. Resource Allocation

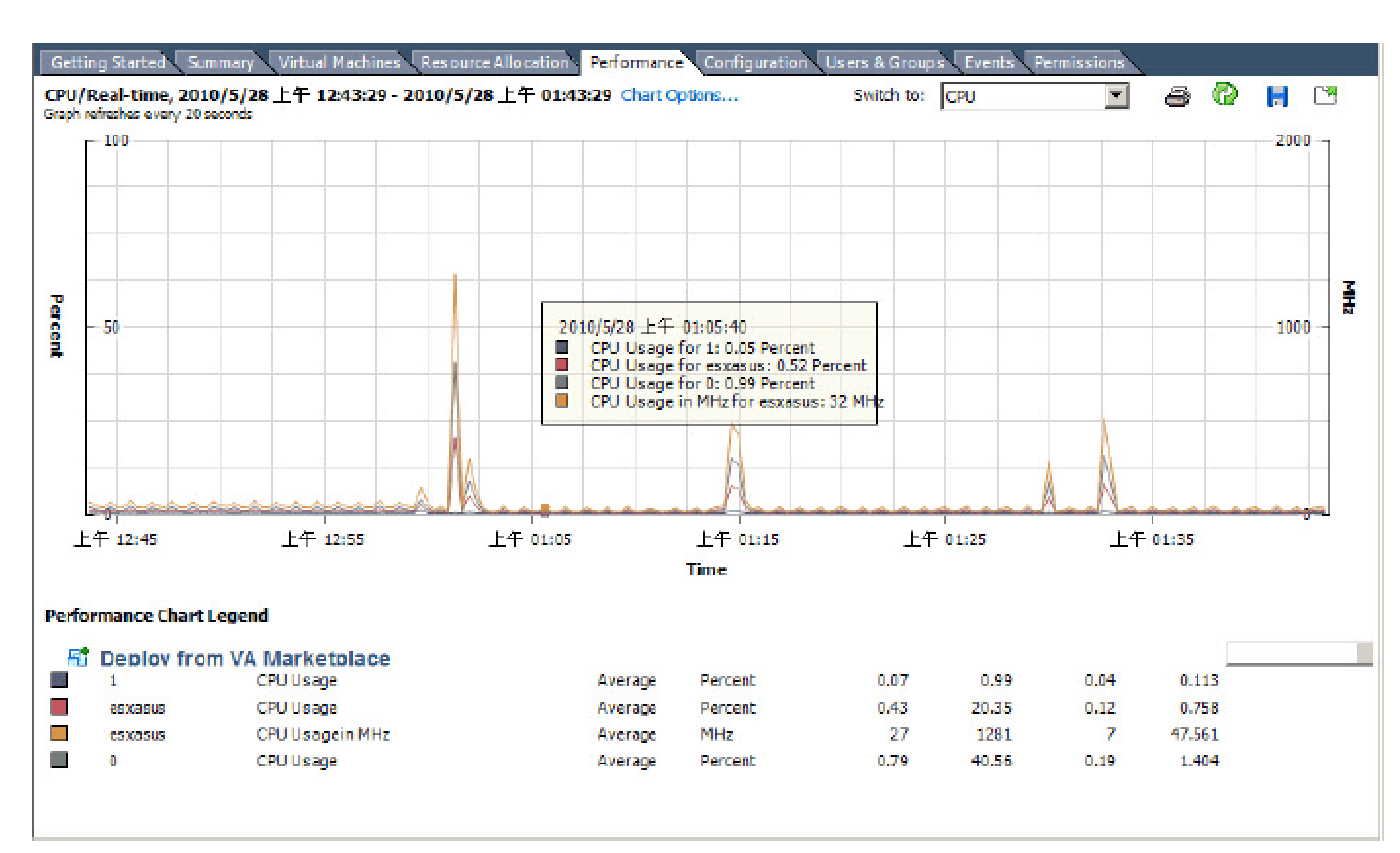
这是在 Host 和 VM 中才会有的选项卡。会显示出当前这个部件的 CPU 和内存的分布情况。我们会在后面的章节说明 CPU 和内存在 vSphere 中的分配。



▲ 资源的分配是 vSphere 整体效能的重要基础

3. Performance

这也是 Datacenter、Cluster、Host、VM 都有的选项卡，也是会显示这个部件的效能图表。这个图表是跟踪效能最重要的功能，如果你发现任何部件在 CPU 或内存上都常常处于满格的情况，就需要重新分配这个部件的 CPU 和内存的资源。举例来说，如果是 VM 的话，必须重新给定更多的 CPU 或内存，如果是 ESX/ESXi 主机，则需要将这个主机上的 VM 移到另一 ESX/ESXi 主机等。



▲ 这是一个性能的图表，是优化 vSphere 的引用数据

结 语

vSphere 中的 vCenter 功能是管理多台 ESX/ESXi 主机的最佳总控台，而理解整个 vCenter 的布局以及使用是今后操作更多高级功能的基础。读者们可以在这一章中自行把玩这些功能，而从下一章开始，我们将陆续介绍多 ESX/ESXi 主机的 vSphere 高级功能。

第 14 章 准备 vSphere 企业实战环境

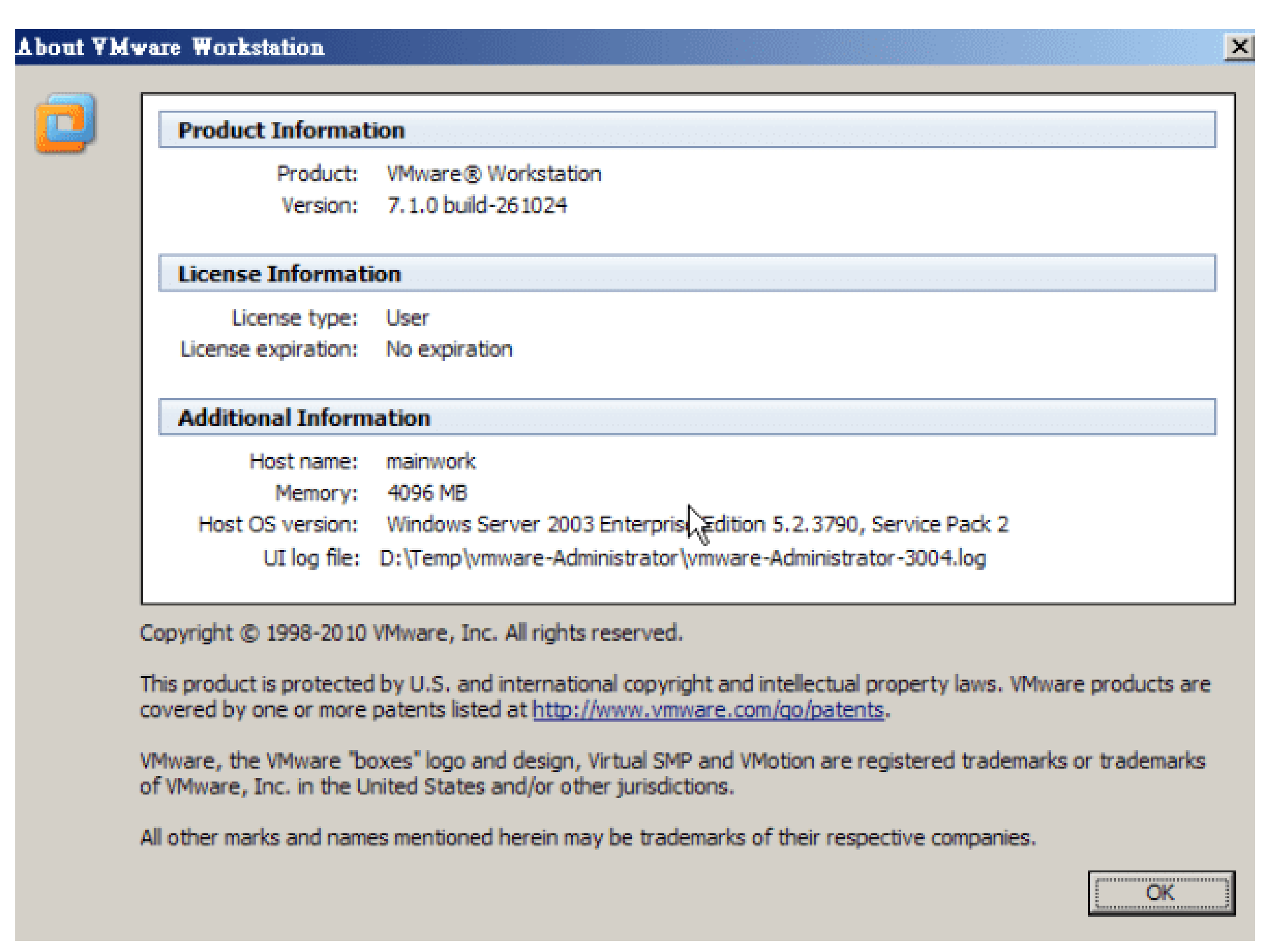
关键词：

- 使用 VMware Workstation 安装 vSphere 环境
- 创建 Host1/2 VM 的网络环境
- 创建 vCenter VM 环境
- 创建 Host 1/2 VM 的 vSwitch 及 Port group
- 创建物理的 vSphere 生产环境
- 理解 vCenter 的部件
- 理解及操作 Template
- 理解及操作 vSphere 中的 Host Profile

前面我们将 vCenter 安装在计算机中，也知道如何将 ESX/ESXi 主机加入 vCenter 中。但 vCenter 的功能非常强大，整个 vSphere 的所有操作都在 vCenter 中进行，因此在接下来的章节中，我们必须架设一个最完整的 vCenter 环境。由于不是每一个读者都有可能准备服务器等级的硬件及存储设备，因此我们将会准备一个实验用环境，当然也会有一个完整的生产环境来作为本书的教材。

14.1 使用 VMware Workstation 7.1 架设实验环境

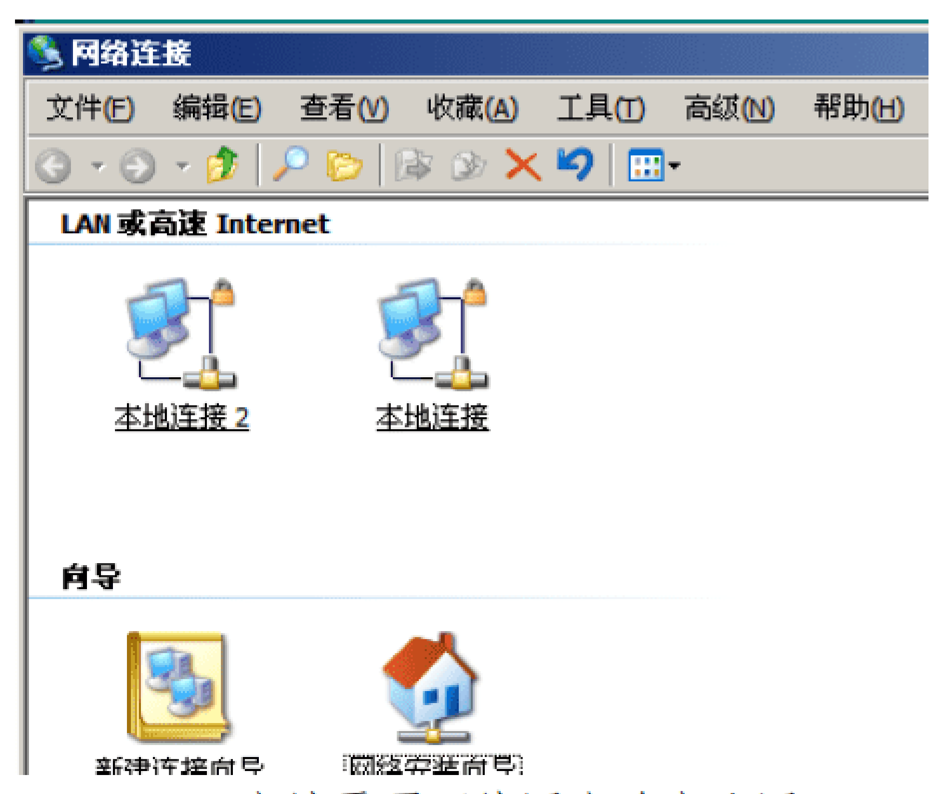
vSphere 至少需要 3 台服务器才能落实整个操作，包括两台 ESX/ESXi 主机及一台安装 vSphere 的 Windows 服务器（2003 或 2008）。但在环境不允许的情况下，我们还是可以利用虚拟机来创建测试环境。VMware 除了 vSphere 之外，也有一个寄居架构的产品，称为 VMware Workstation。VMware Workstation 必须创建在一个已安装好的操作系统中，但因兼容性较高，并且也支持在 VM 中安装 ESX/ESXi，因此经常用在测试环境中。要使用 VMware Workstation 来安装 vSphere 的测试环境，下面是必须的条件：



▲ 当前 VMware Workstation 是最广泛使用的产品

▶ **使用 VMware Workstation 运行 vSphere 环境的条件**

- 支持 VT 或 AMD-V 技术的 CPU。
- Windows XP 以上的操作系统或具有 X-Windows 的 Linux 操作系统。
- 4GB 以上的内存。
- 两片网卡，一片连上公网，一片连上存储设备（iSCSI 或是 NFS）。



▲ 本地需要两片网卡连上公网

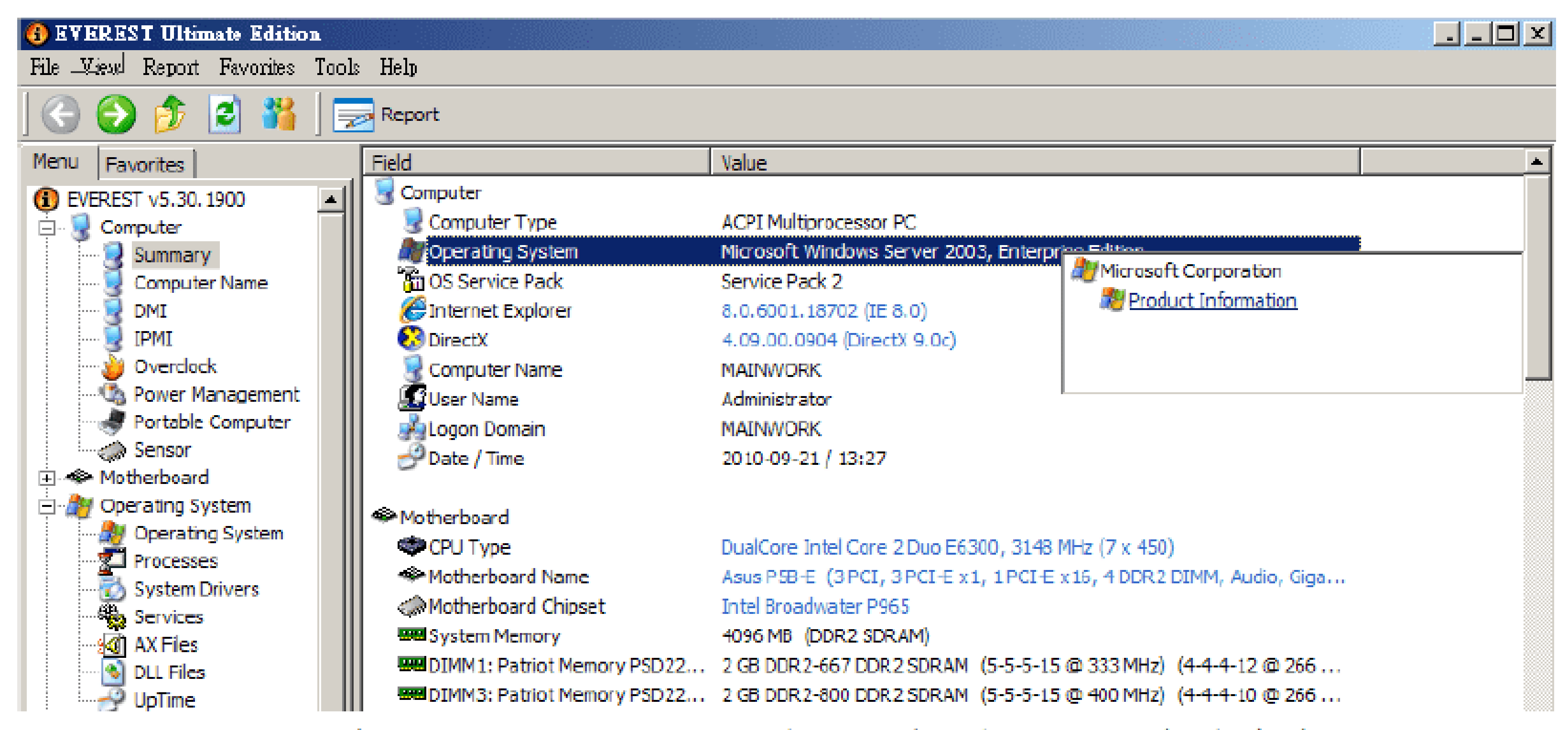
当你的设备准备好之后，我们就来看看如何在 VMware Workstation 中创建一个 vSphere 的实验环境。

14.1.1 下载并安装 VMware Workstation 7.1

当前 VMware Workstation 最新的版本是 7.1，下面是本地测试的环境。

▶ **本书 vSphere 测试环境**

- 操作系统：Windows 7 Ultimate X64 版本
- CPU：Intel Duo Core2 E6300 3.16GHz 双核
- 内存：4GB
- 硬盘：320GB SATA2
- 网卡：Intel 1000MT 网卡 2 片，一片连上公网，一片连上 iSCSI 设备
- 网络 IP：192.168.1.0
- iSCSI：10.1.1.0



▲ 在 Windows 2003 下，也可以抓到 4GB 以上的内存

1. 下载 VMware Workstation 7.1

你可以到 VMware 的官方网站下载 30 天试用版，或是直接购买 VMware Workstation 产品。当前最新的版本是 7.1。当你准备好兼容的硬件之后，就可以安装 VMware Workstation 了。下载地址为：http://downloads.vmware.com/d/info/desktop_downloads/vmware_workstation/7_0。

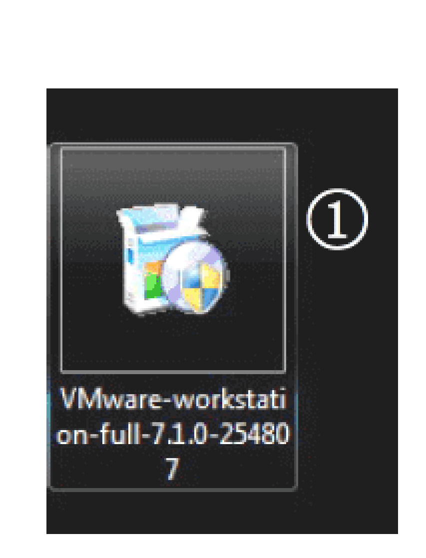


▲ 先来这里下载试用版，可以用 30 天

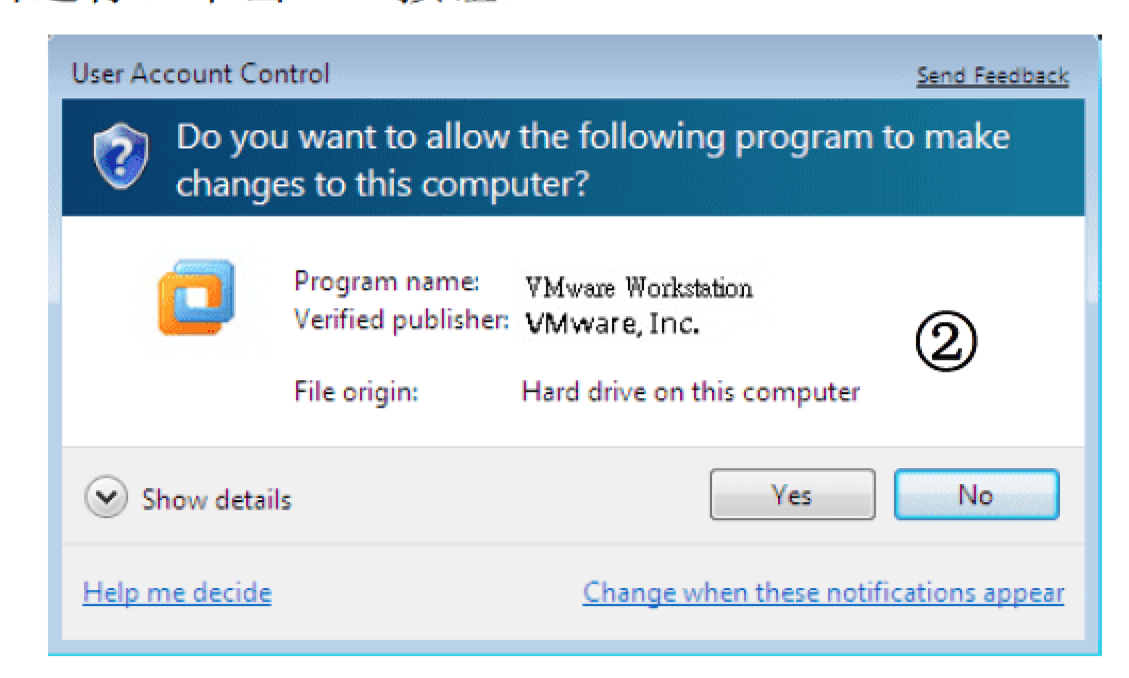
2. 安装 VMware Workstation 7.1

下载回来 VMware Workstation 的运行文件之后就可以安装了，下面就是步骤。

1. 直接运行 VMware Workstation 的运行文件。
2. 此时系统会让你再确定是否运行。单击 Yes 按钮。

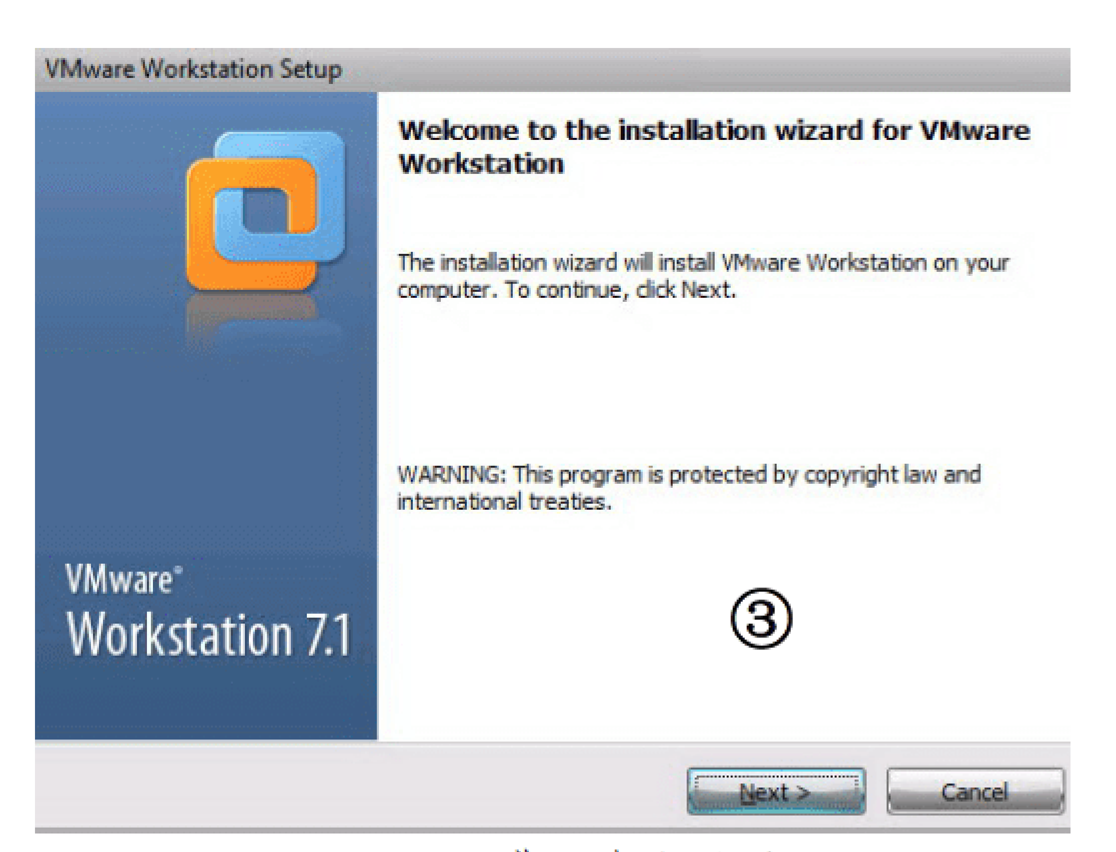


▲ 运行下载回来的运行文件

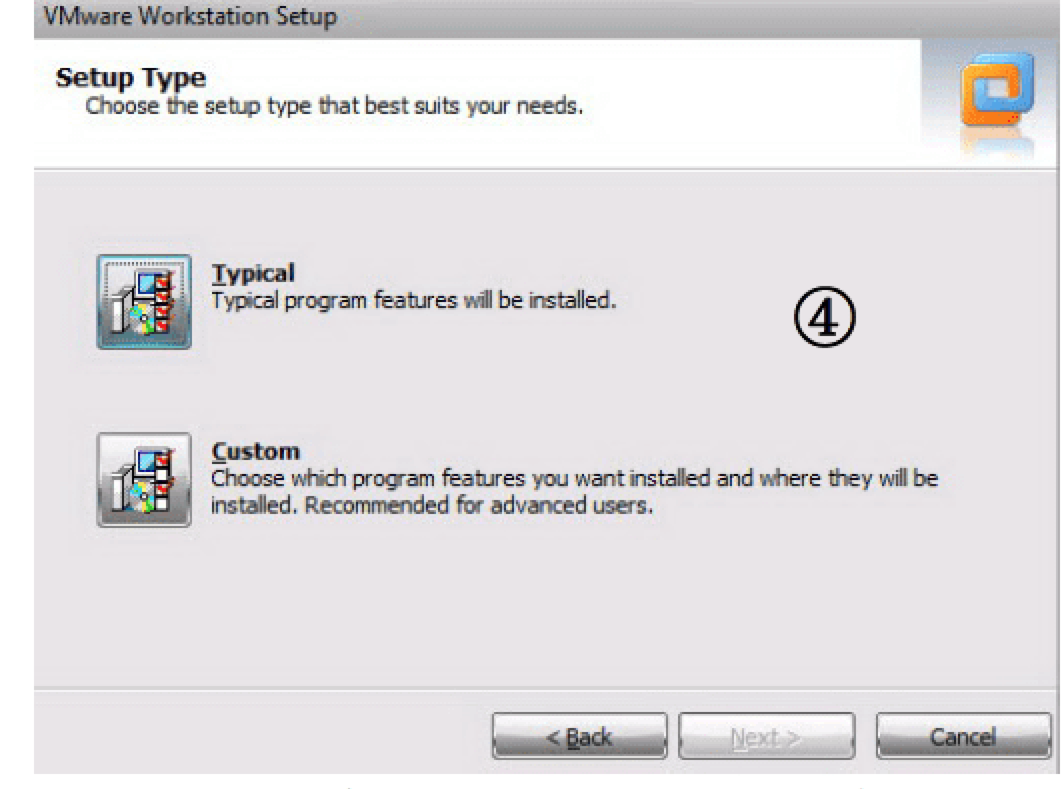


▲ 在 Windows 7 下必须先通过用户总控的功能

3. 系统进入安装解压画面。当弹出如下窗口时，单击 Next 按钮。
4. 选择 Custom 进行自定义安装。

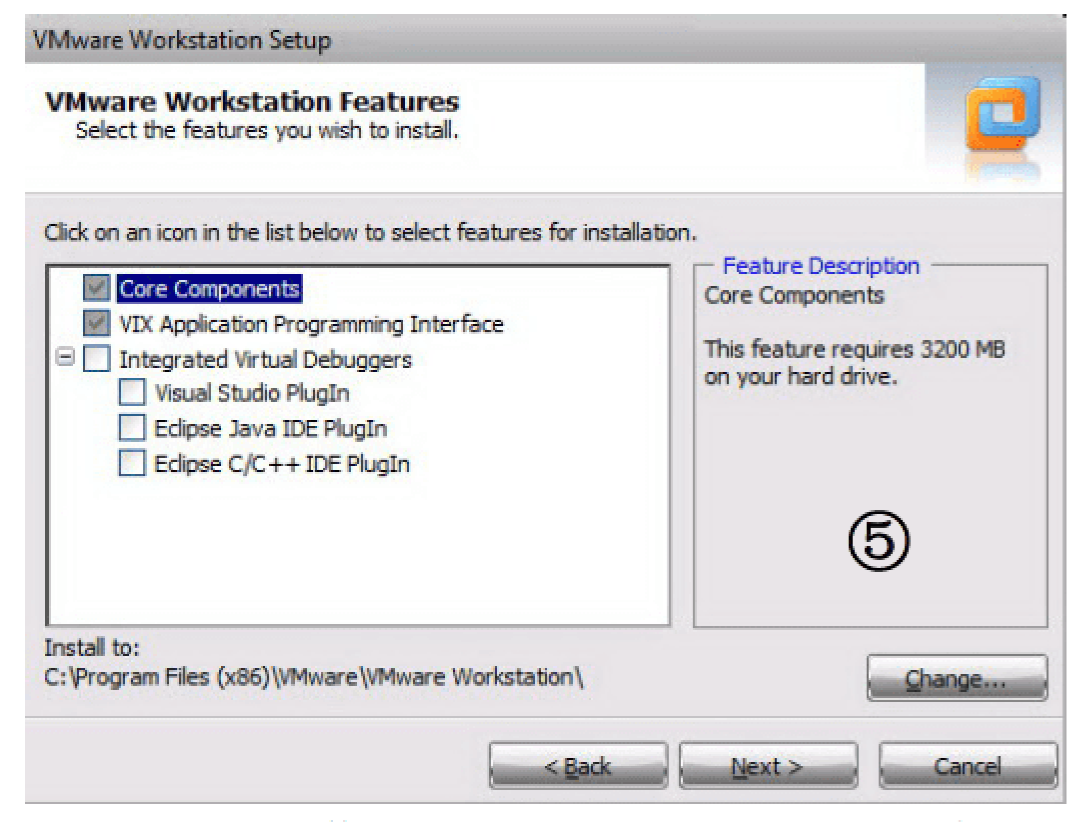


▲ 进入欢迎画面

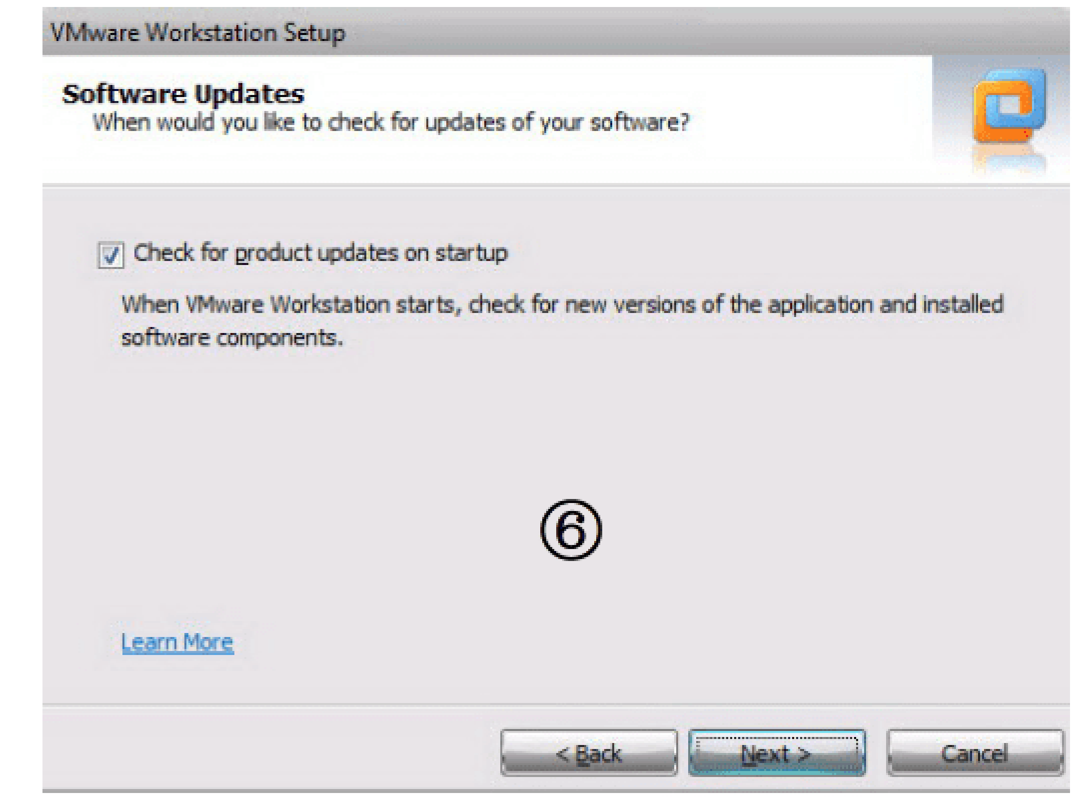


▲ 进入自定义安装的选项较多

5. 在这个窗口中可以什么都不选，直接单击 Next 按钮。
6. 在勾选是否要遍历产品补丁的复选框之后，单击 Next 按钮。



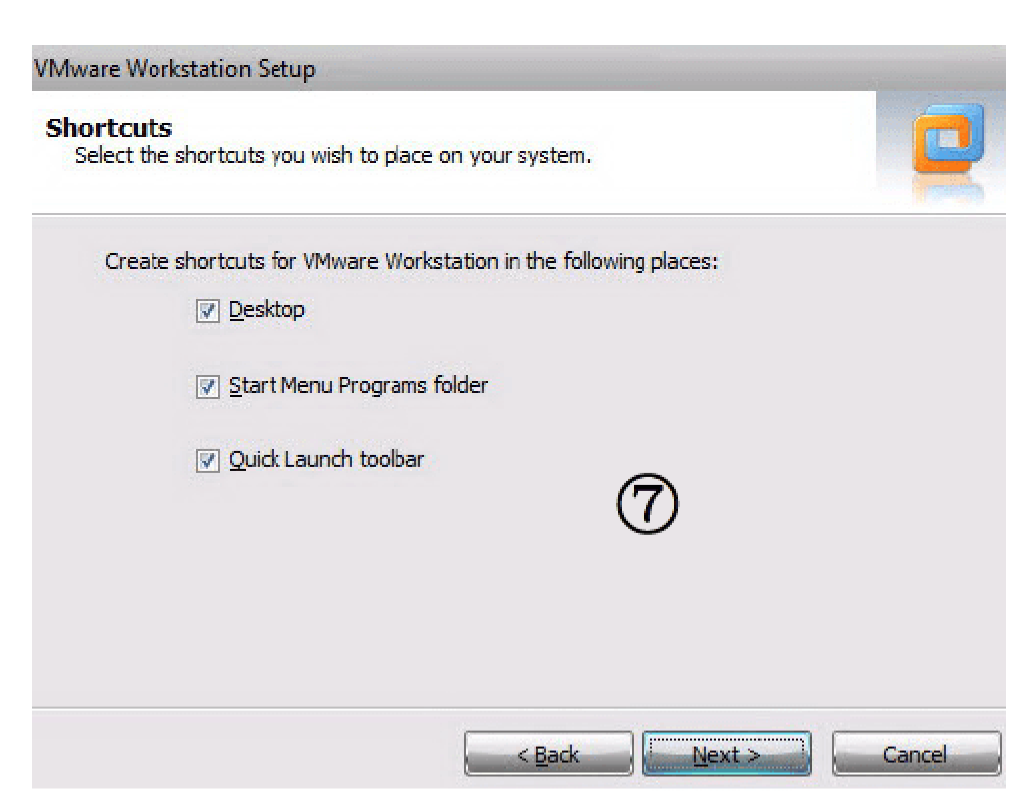
▲ 如果你不打算在 VMware Workstation 下开发，就直接到下一步



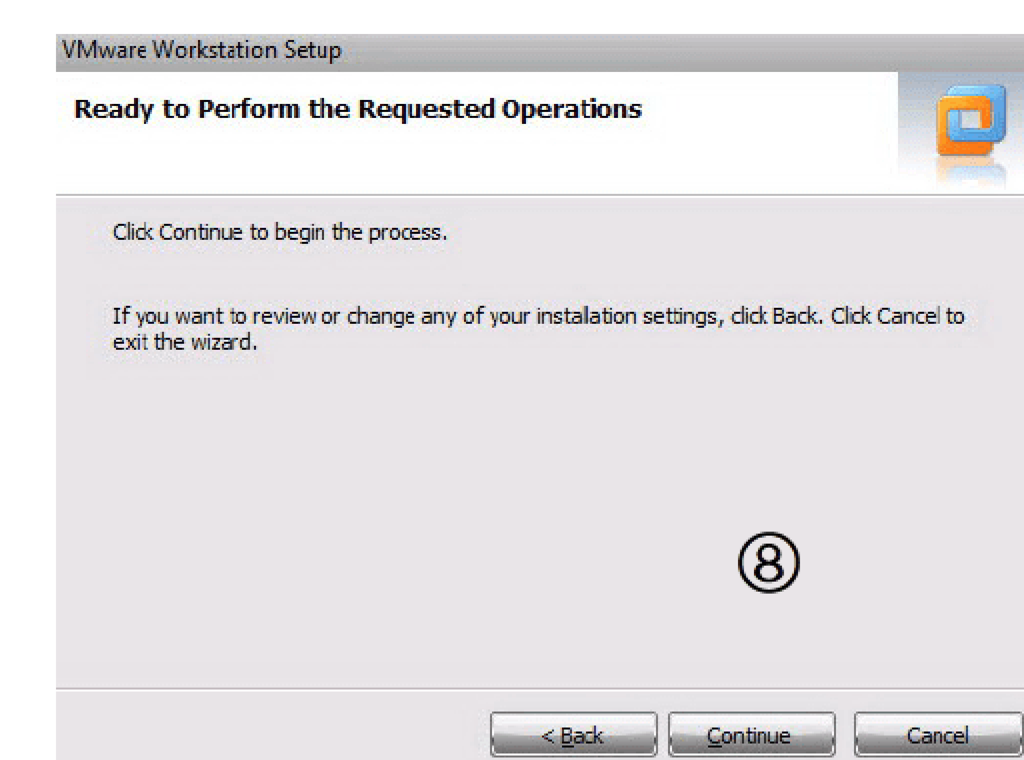
▲ 可以遍历软件补丁

7. 接下来是否在桌面等地方创建快捷方式的地方，也单击 Next 按钮。

8. 接下来是总结画面，单击 Continue 按钮。



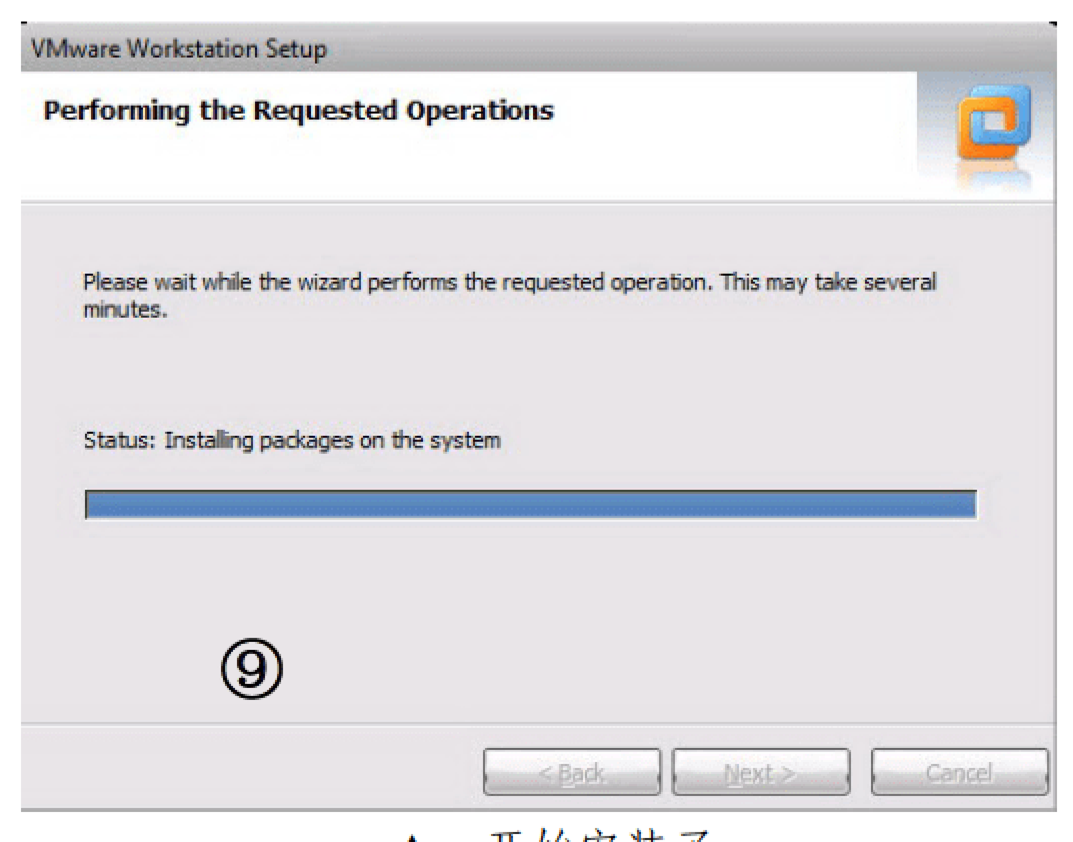
▲ 创建桌面快捷方式



▲ 总结画面

9. 接下来系统会开始安装，会弹出安装画面。

10. 当弹出如下图所示画面时，键入序号。

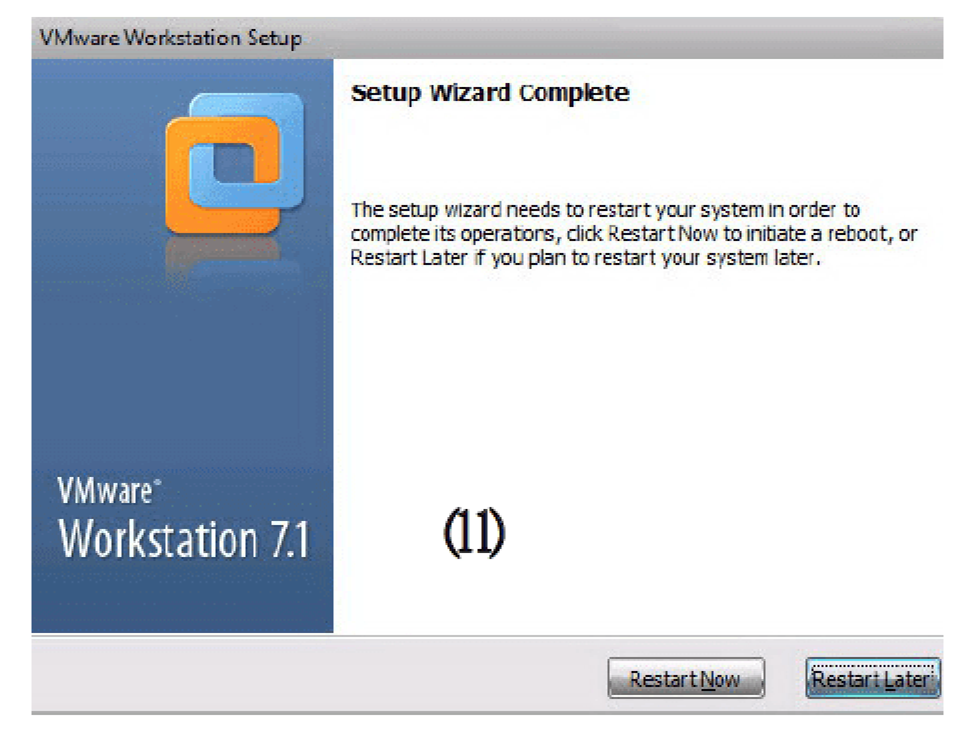


▲ 开始安装了

▲ 键入序号，试用版就单击 Skip 按钮

11. 当弹出如下图所示画面时，意味着已经安装完毕，单击 Restart Now 按钮重新激活后即可以使用。

12. 当重启之后，就可以看到已经在桌面上弹出 VMware 的图示了。



▲ 重启后就可以使用了

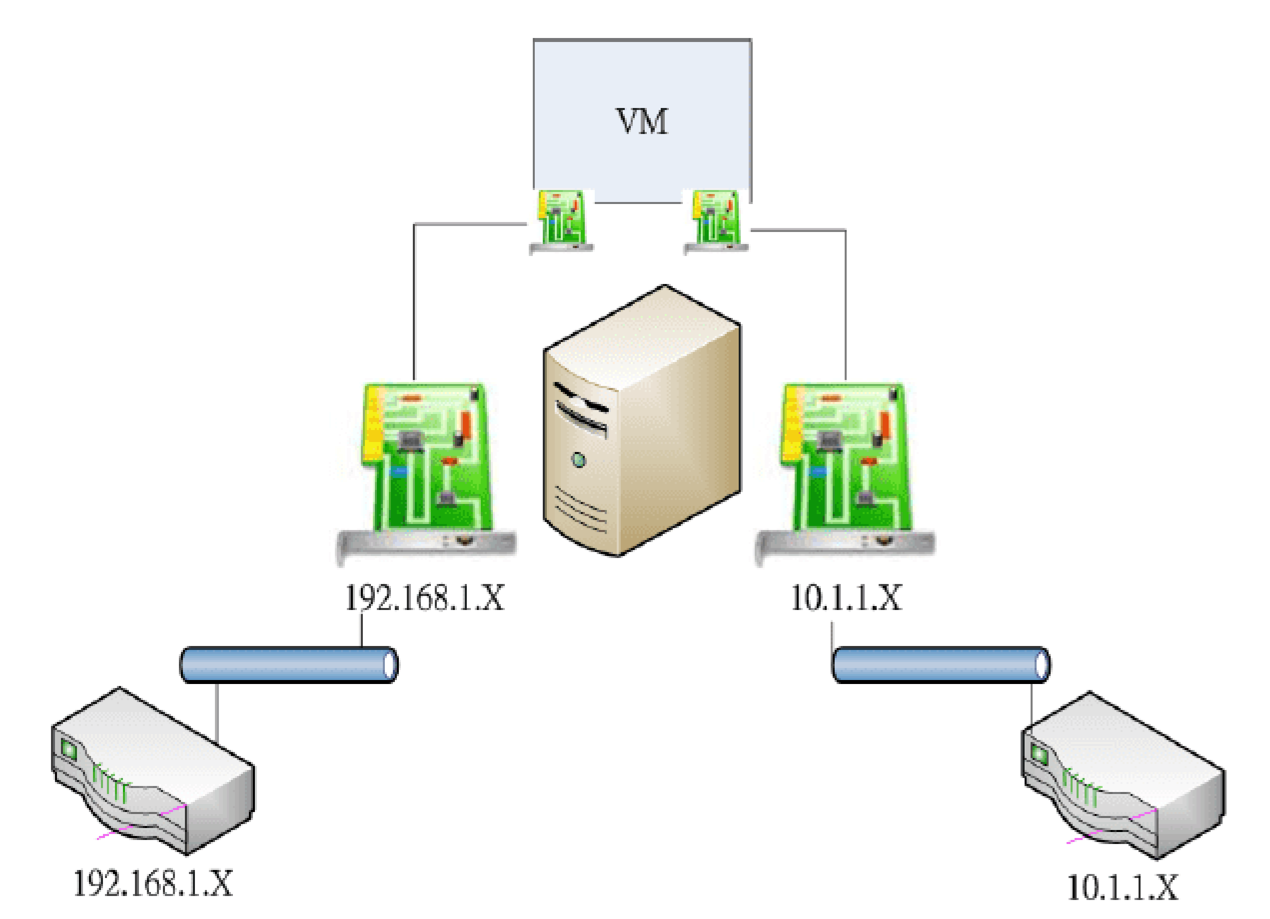


▲ 这就是大名鼎鼎的 VMware Workstation

在安装好 VMware Workstation 后，你的计算机就可以安装 vSphere 环境了。VMware Workstation 7.1 会将操作系统所掌管的硬件资源再重新分配，接下来只要在原来的主操作系统上创建 ESX/ESXi 及 vCenter 的 Windows 2003 虚拟机即可。

14.1.2　改变 Vmware Workstation 7.1 的网络配置

在 VMware Workstation 下，网络配置和 ESX/ESXi 并不相同。VMware Workstation 主要是以使用主系统的网卡来连上物理路由器，也可以选择其他的网络配置。由于我们的总控机器和存储设备都是物理网卡，因此必须将 VMware Workstation 中 VM 的网卡再桥接至物理网卡。

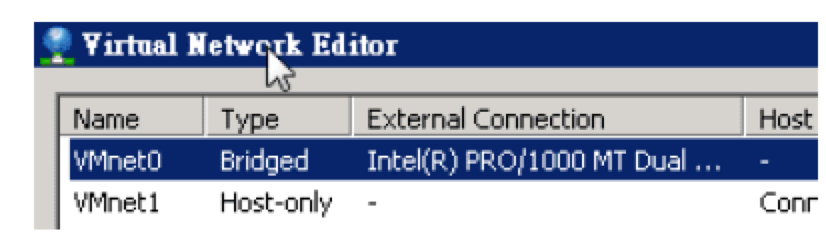


▲　两片网卡连上不同的网段，也要确保 VM 能连上这两个网段

1. 理解 VMware Workstation 的网络配置

VMware Workstation 提供了许多网络联机方式，和 ESX/ESXi 不同的是，VMware Workstation 使用虚拟路由器 VMnetx，而 ESX/ESXi 使用的是 vSwitch 上的 Port Group。但 VMware Workstation 的虚拟路由器只有 9 个，并且在 IP 的配置上也受限于主系统的网卡，因此在配置时必须遵守其原则。

要让 VMware Workstation 上的 VM 能通过其虚拟路由器连上公网，必须使用 Bridged 模式。要让 Bridged 模式生效，必须在 VMware Workstation 的网络配置上创建桥接的目标，如此一来，在 VMware Workstation 中的 VM，也可以和主系统以及其他连上这个网段的物理计算机或虚拟机之间成为同一层级的网络交流。



▲　默认已经使用一个桥接了

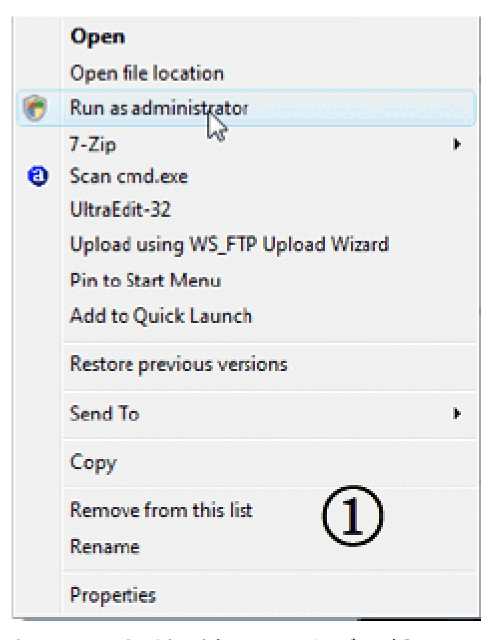
2. 配置 VMware Workstation 的多重桥接网络

由于物理机有多片网卡，因此在 VMware Workstation 的 VM 也可以配置两片以上虚拟网卡，并且分别将这两片网卡桥接到不同的物理网卡，以达到 VM 中的 ESX/ESXi 也可以连上总控台和存储的公网。下面就是 VMware Workstation 中的网络配置。

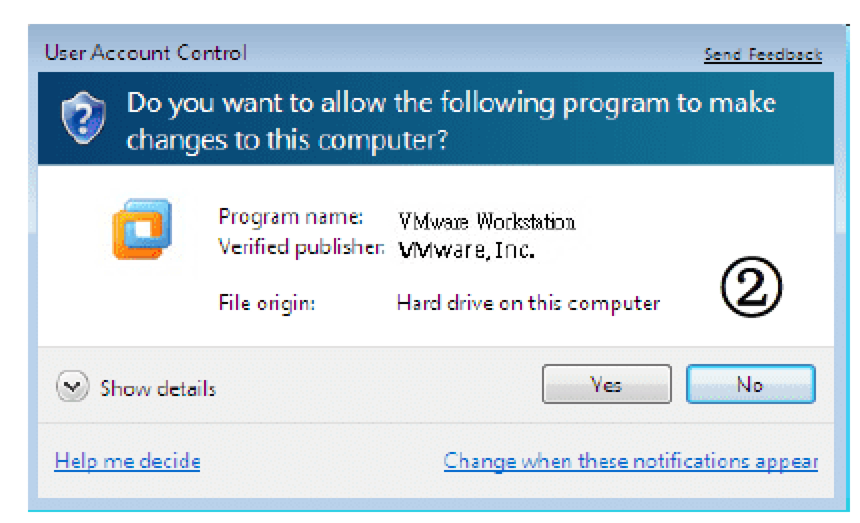
3. 配置 VMware Workstation 的桥接配置

1．在 Windows 7 下运行 VMware Workstation，但不要直接运行，必须使用系统管理员的身份才能改变网络配置。在 VMware Workstation 的图标右击，选择系统管理员的身份运行。

2．此时会弹出 Windows 7 的用户安全画面，单击 Yes 按钮。



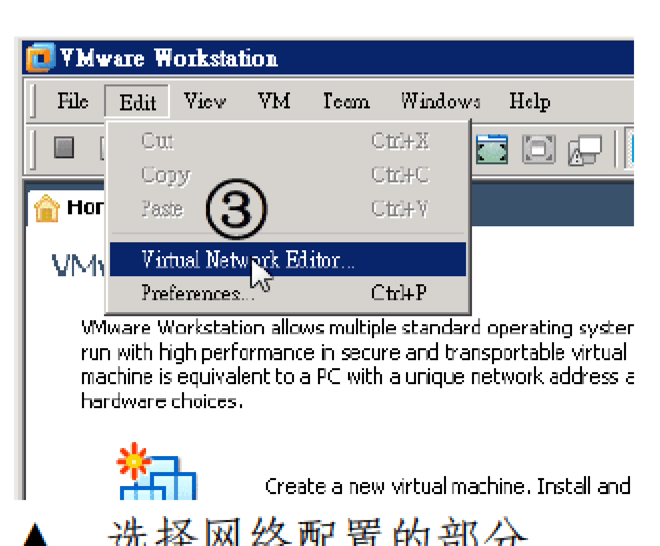
▲ 必须用系统管理员身份运行



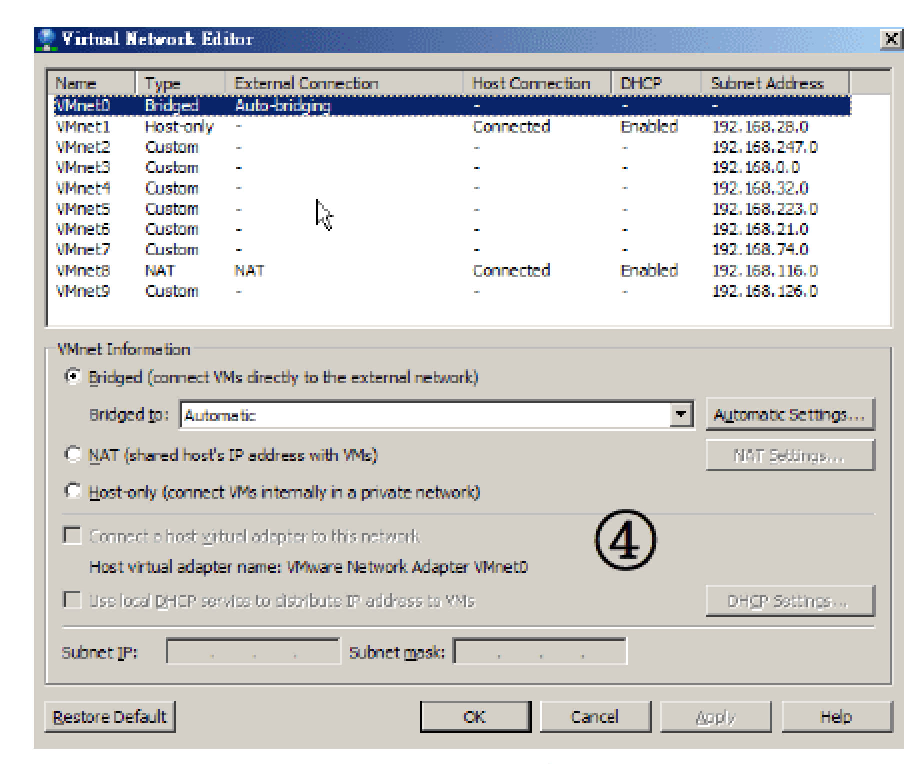
▲ 单击 Yes 按钮

3．此时会进入 VMware Workstation 的画面。选择 Edit/Virtual Network Editor 选项。

4．在没有更改过 VMware Workstation 的网络配置之前，系统的默认值就是将第一个网卡自动桥接到物理网卡上。但我们这里要自行配置桥接选项，因此先单击 Restore Default 按钮。



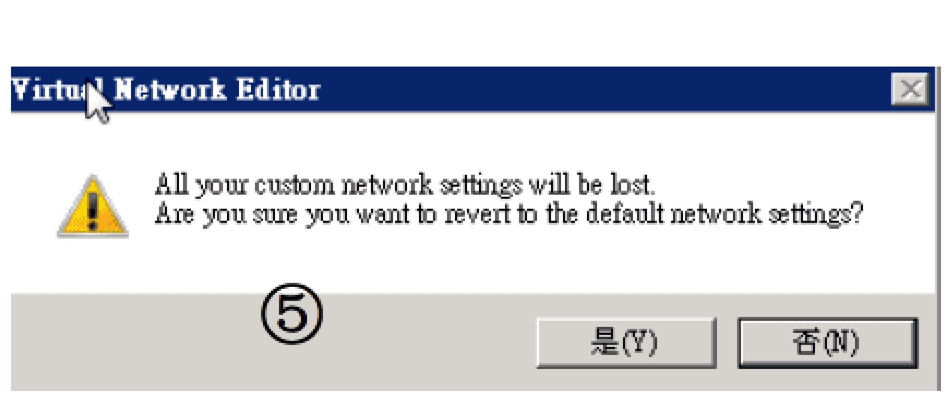
▲ 选择网络配置的部分



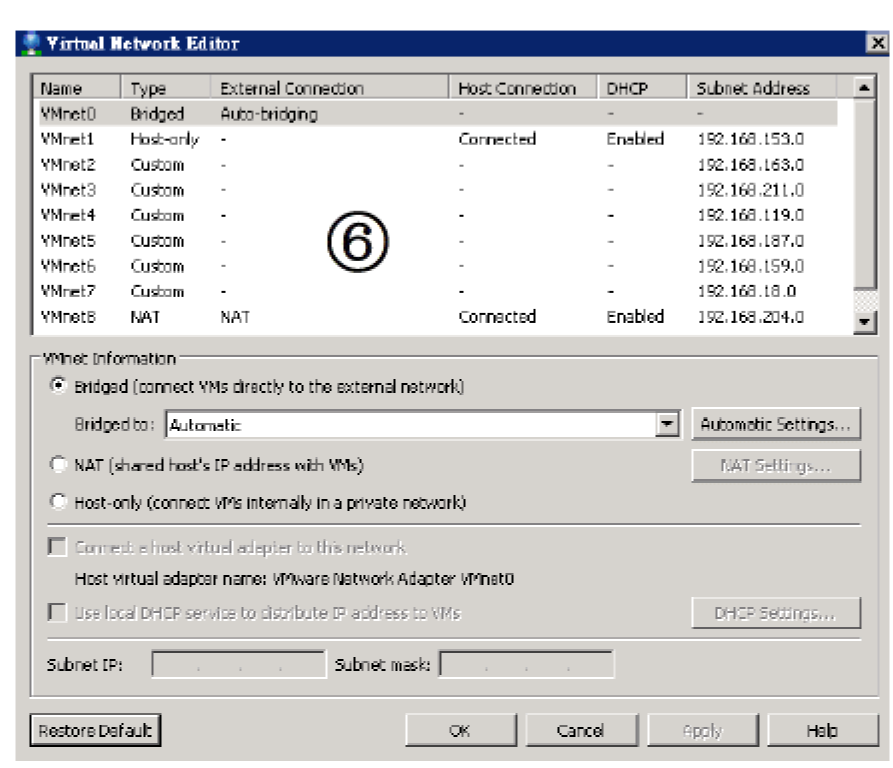
▲ 先将网络配置重设

5．系统会开始重新配置网卡，会先有报警消息。

6．接下来就会开始重新配置，将会花上几分钟。配置完毕之后，我们可以看到在 VMware Workstation 的不同网卡已经对应完毕。其中第一个虚拟路由器 VMnet0 是自动桥接的，一共有 9 个虚拟路由器。

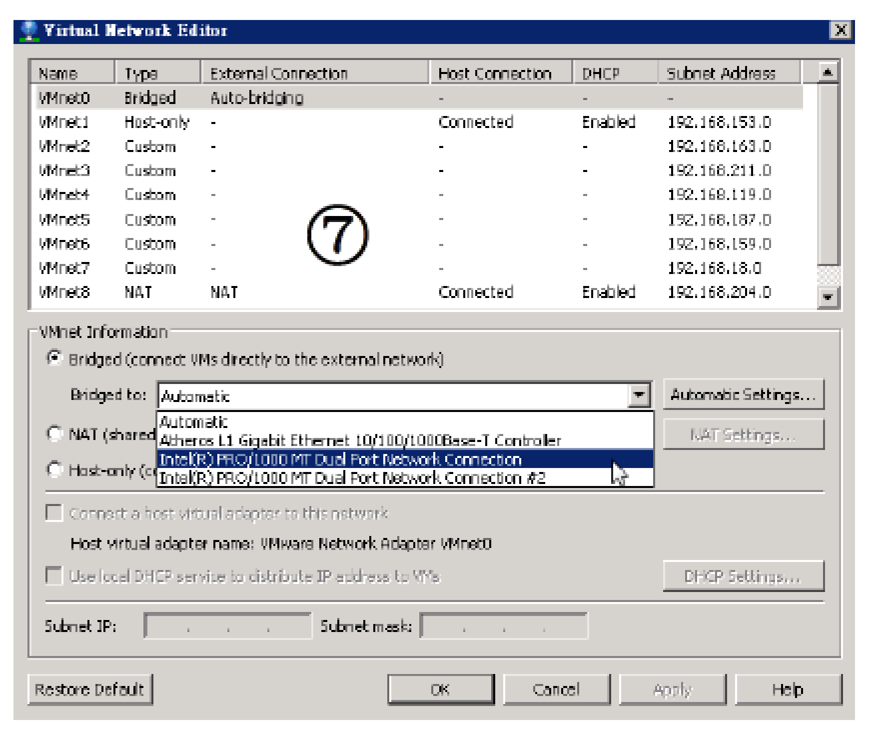


▲ 开始重设网卡



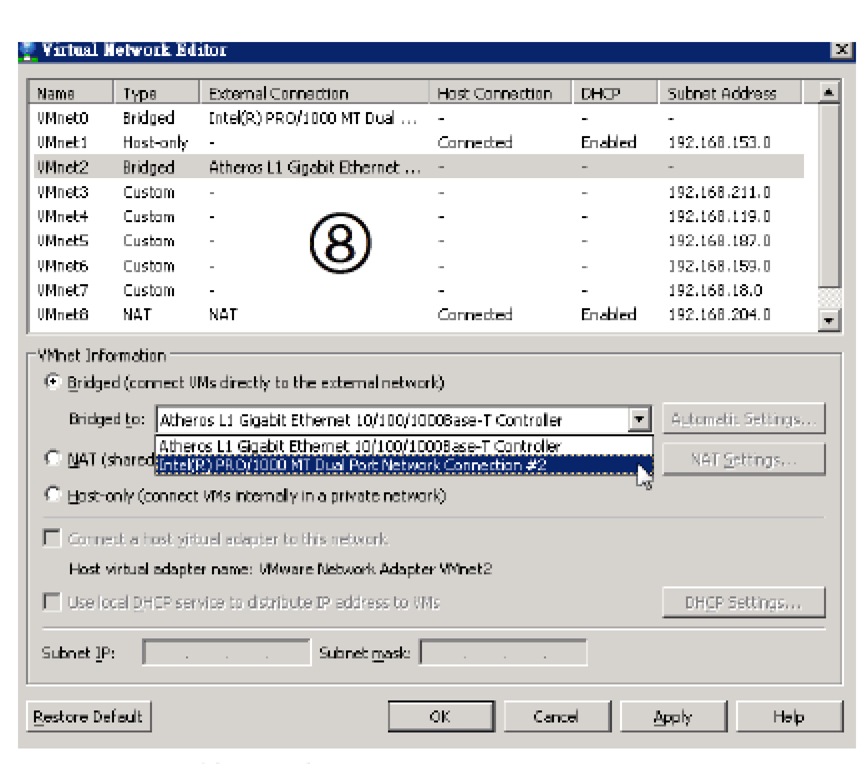
▲ 这是默认值，我们要将某一个虚拟网卡桥接到第二张网卡

7．选中第一个虚拟路由器 VMnet0，并单击 Bridged 单选按钮，但是在 Bridged to 列表框不要选择 Automatic，而是连上公网的第一片 Intel 网卡。这个网卡是连 192.168.1.X 这个公网。



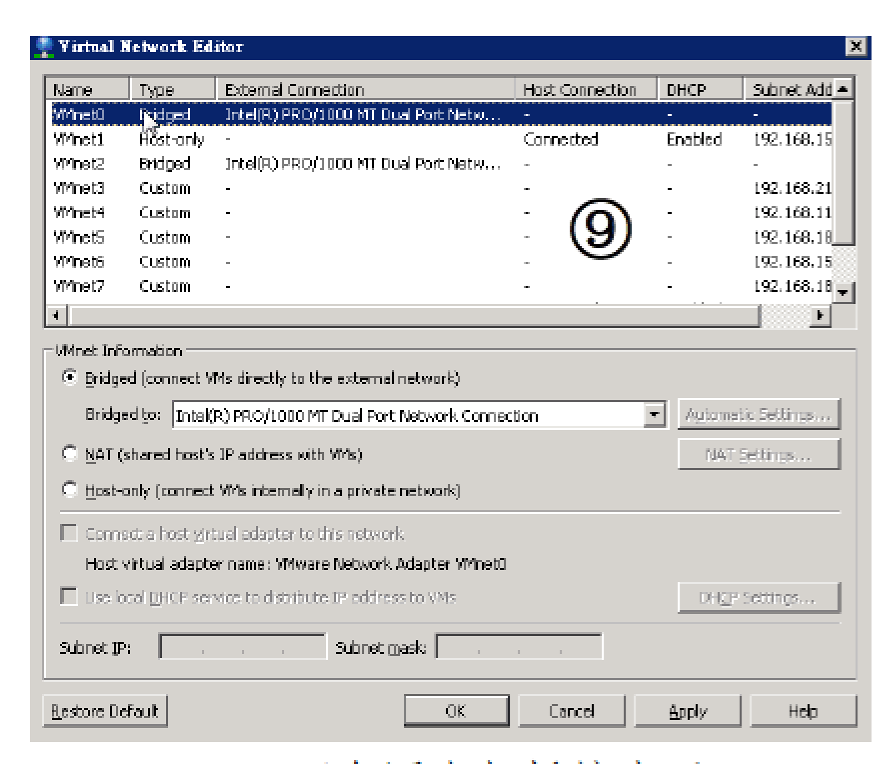
▲　先将第一张网卡的桥接配置好，不要用 Auto

8．由于 VMnet1 已经被用了，因此我们将 VMnet2 配置成第二个桥接网络，桥接上 10.1.1.X 这个存储公网。选中 VMnet2，并单击 Bridged 单选按钮，并且在 Bridged to 列表框选择第二片 Intel 网卡。刚才桥接到 192.168.1.X 的物理网卡已经被用掉，因此不会弹出在清单中。



▲　将第二片物理网卡给定到 VMnet2

9．配置完毕后，可以看到两个网段都已经可桥接了，意味着在这台计算机 VMware Workstation 上的 VM，也可以连接到这两个物理网卡上。接下来单击 OK 按钮即可。



▲　两个网卡都桥接上了

14.2 在 VMware Workstation 中安装 vSphere 环境

在安装好 VMware Workstation 及网络配置之后，接下来就是安装 ESX/ESXi 的 VM 了。

14.2.1 配置网络环境

一个完整的 vSphere 需要至少三台机器，包括两台 ESX/ESXi 和一台 vCenter 的 Windows 2003 Server。在开始之前，我们先来将这几台计算机的 IP 及网络配置定义下来。

1. 定义 IP 及主机名

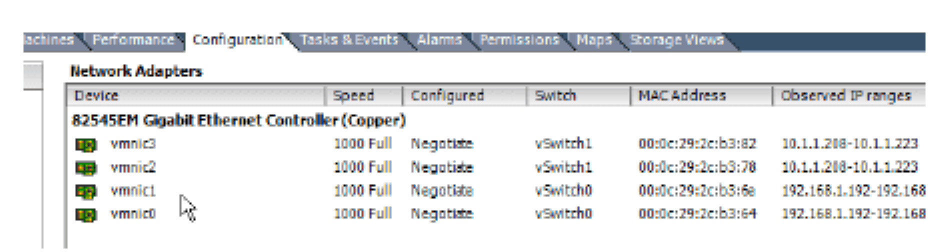
在 vSphere 中可以使用 IP 加入主机，但缺点是如果使用 IP 的话，在 vSphere Client 中的 Performance 选项卡会看不到数据，因此除了 IP 之外，也必须定主机名。

vMware Workstation 中的 vSphere 基本环境

vCenter Server	192.168.1.243
iSCSI 主机	10.1.1.223/192.168.1.223
ESX host1	192.168.1.81
ESX host2	192.168.1.82
ESX host1 iSCSI	10.1.1.81
ESX host2 iSCSI	10.1.1.82
ESX host1 主机	host1.vaemon.com
ESX host2 主机	host2.vaemon.com

2. 网卡选项

在 VMware Workstation 中的 VM 可以有多片网卡。我们可以将 ESX/ESXi 的 VM 配置多片网卡，在稍后章节的 NIC Team、Fault Tolerance、HA 等功能演示上较为方便，因此我们的 ESX/ESXi VM 网卡配置是，每一个网段都创建两片虚拟网卡。



▲ 一个 ESX/ESXi 主机通常需要多片网卡

在 VMware Workstation 下创建 VM 和在 ESX/ESXi 下创建大同小异，只要注意网卡的桥接段即可。下面就分别来看看创建不同虚拟机的方式。

14.2.2 创建 vCenter 的 Windows 2003 Server

首先我们必须创建一台 VM，并且在创建完之后将 VMware vCenter 安装起来。

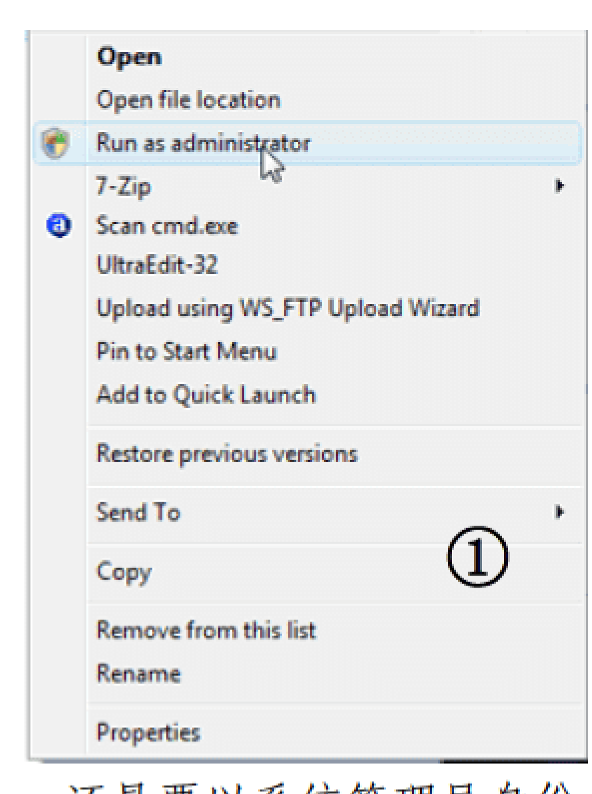
1. 创建 Windows 2003 Server 的虚拟机

VMware vCenter 所需要的资源并不算多，我们只要能运行 vCenter 服务即可。下面就是在 VMware Workstation 下创建 VM 的完整步骤。

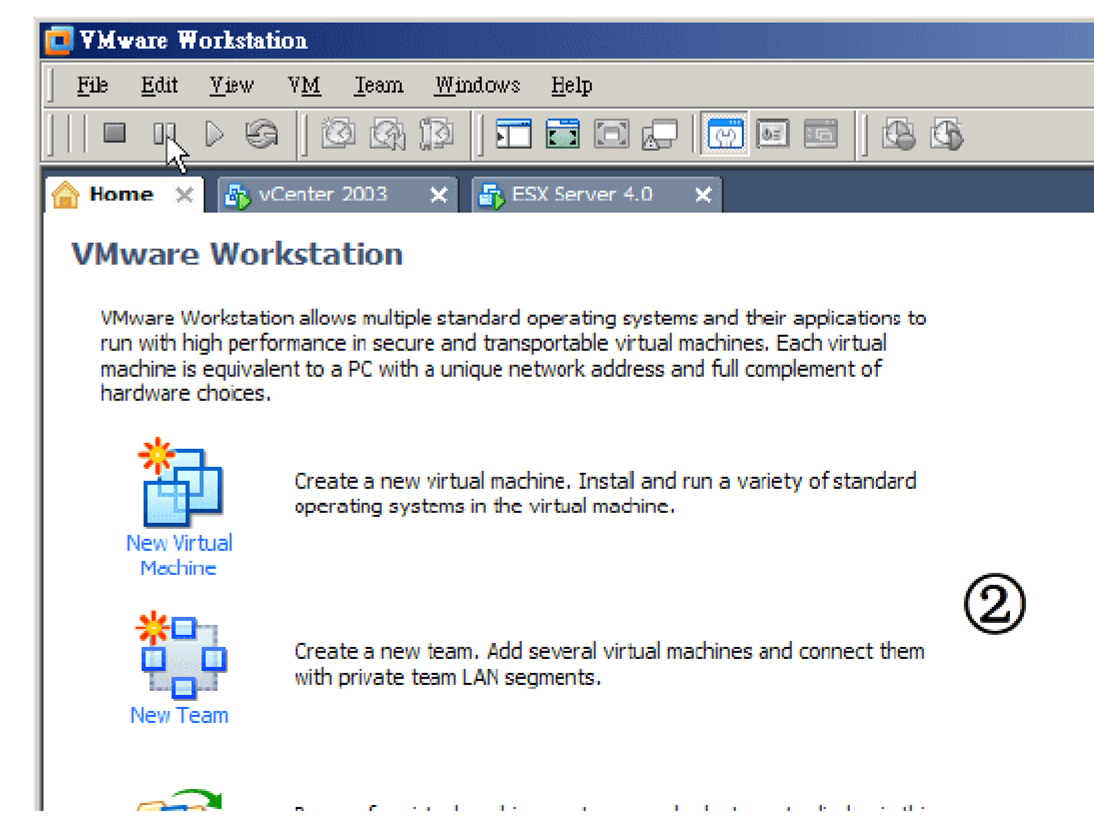
▶ 在 VMware Workstation 下创建 VM

1. 还是以系统管理员的身份进入 VMware Workstation 中。

2. 选择 New Virtual Machine。



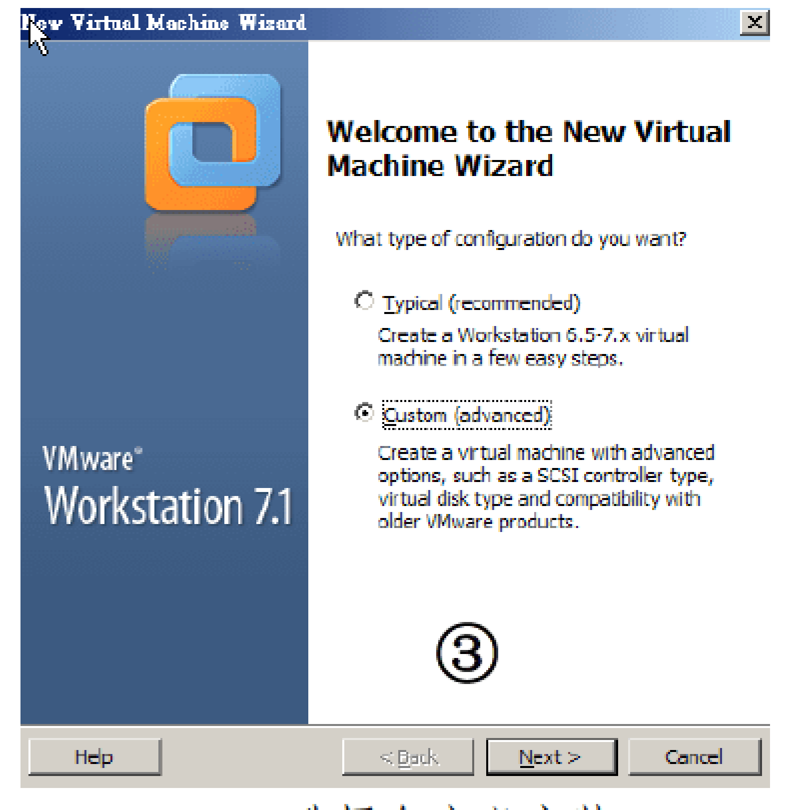
▲ 还是要以系统管理员身份



▲ 选择新建 VM

3. 接下来是安装选项，单击 Custom 单选按钮，单击 Next 按钮继续。

4. 选择 VMware Workstation 的硬件兼容性，这里选择第一项即可，单击 Next 按钮继续。

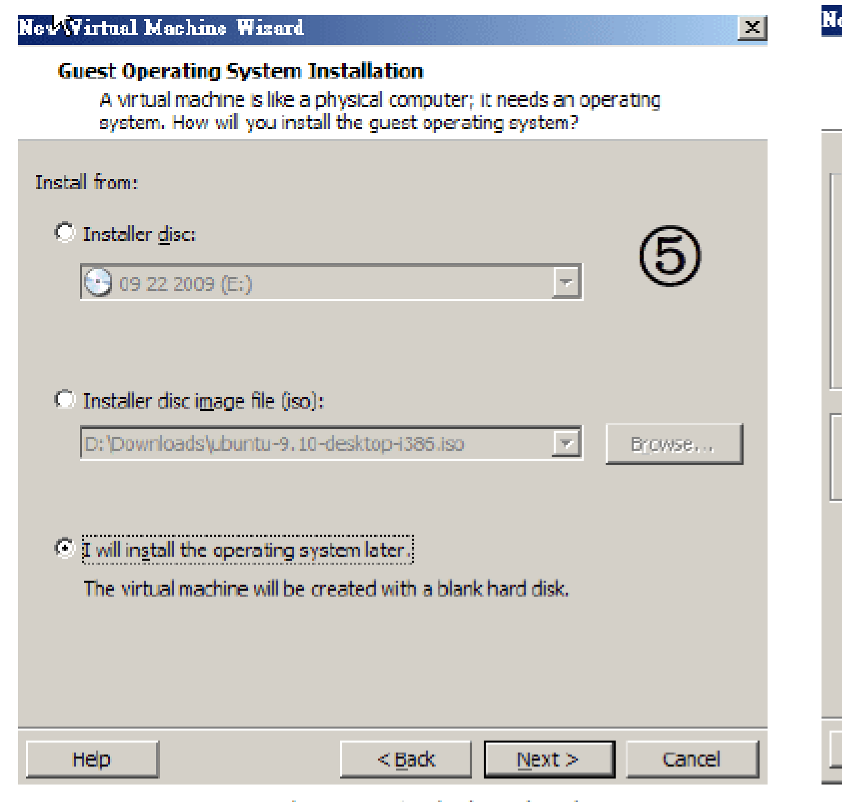


▲ 选择自定义安装

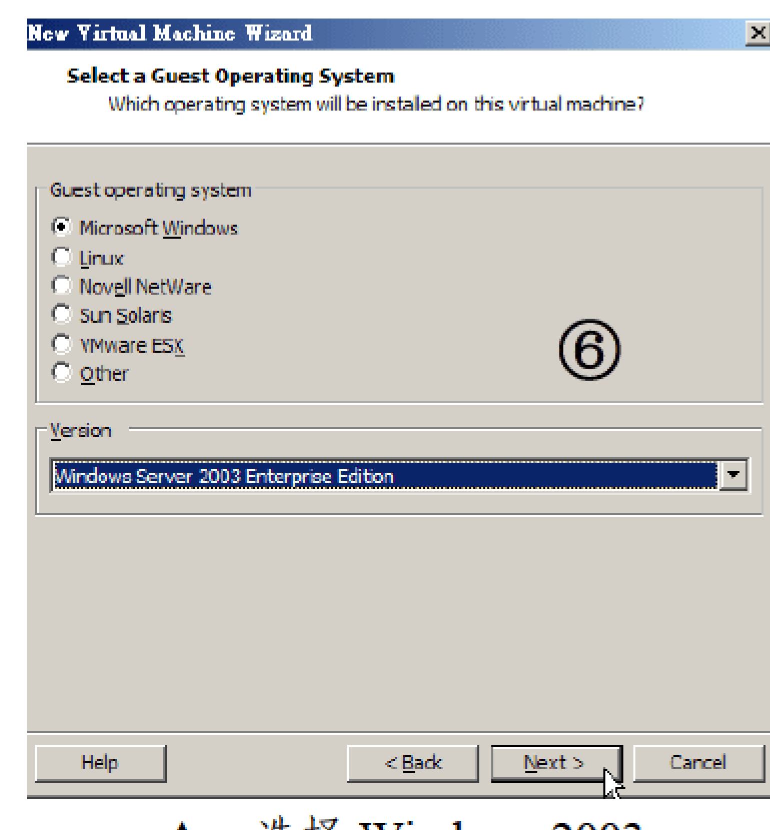
▲ 选择第一项

5. 光盘机选择，当前先不要选，单击 Next 按钮继续。

6. 接下来是选择安装的操作系统，选择上面的 Windows，下面选择 Windows 2003。



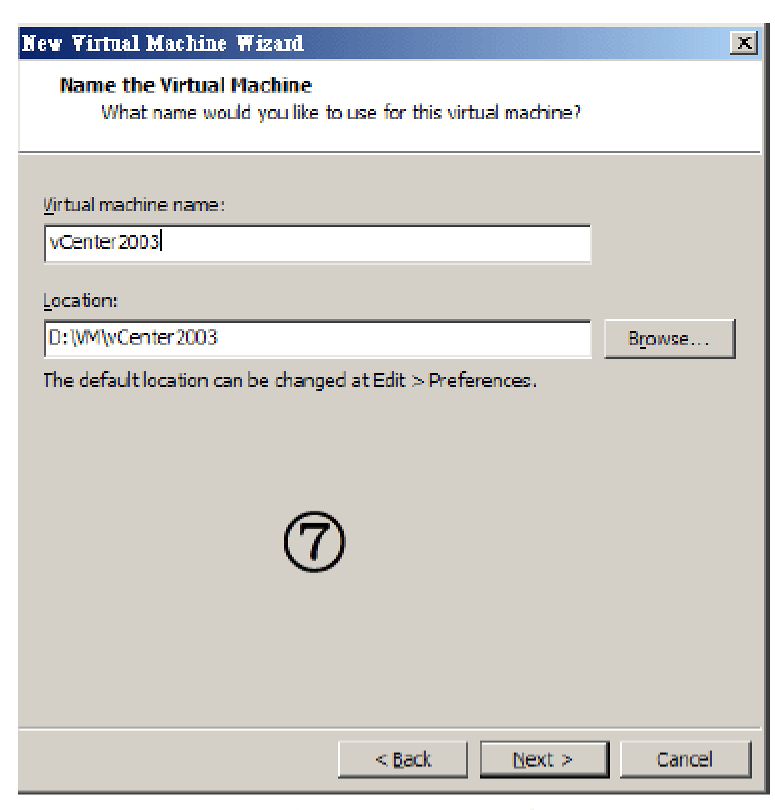
▲ 先不要选择光盘

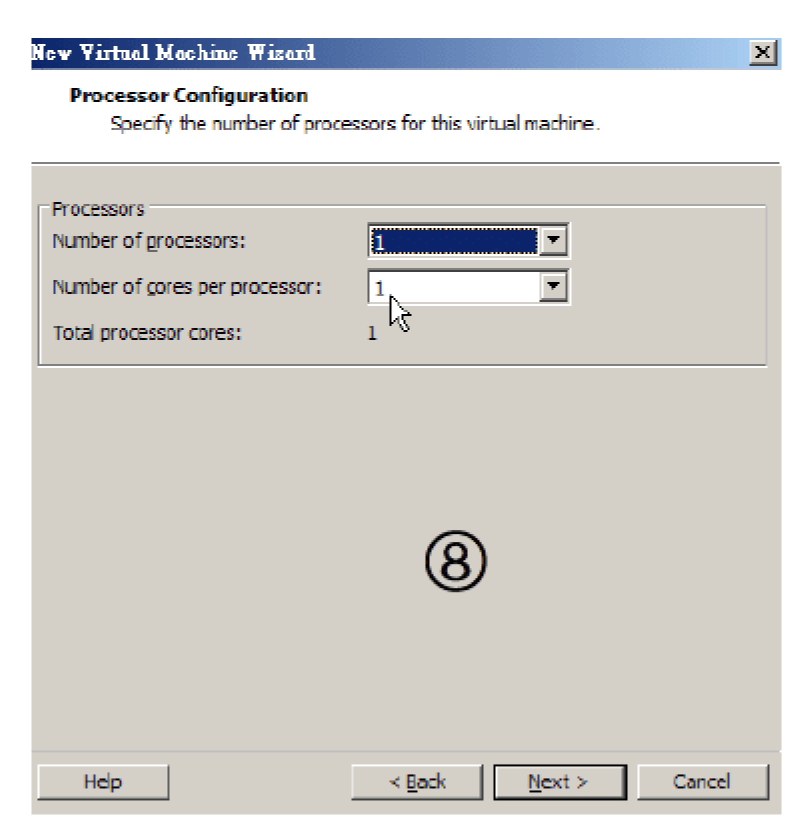


▲ 选择 Windows 2003

7. 接下来是替这台 VM 命名，我们就键入 vCenter 2003，下方是这台 VM 存储的地点。单击 Next 按钮继续。

8. 接下来是选择这个 VM 的 CPU，就选择一个即可，单击 Next 按钮继续。

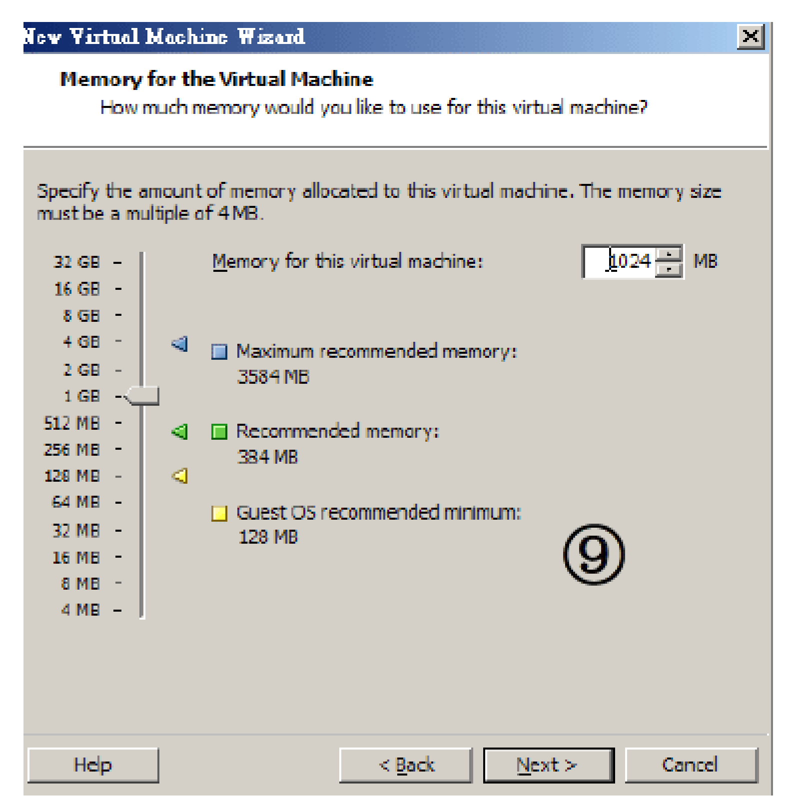


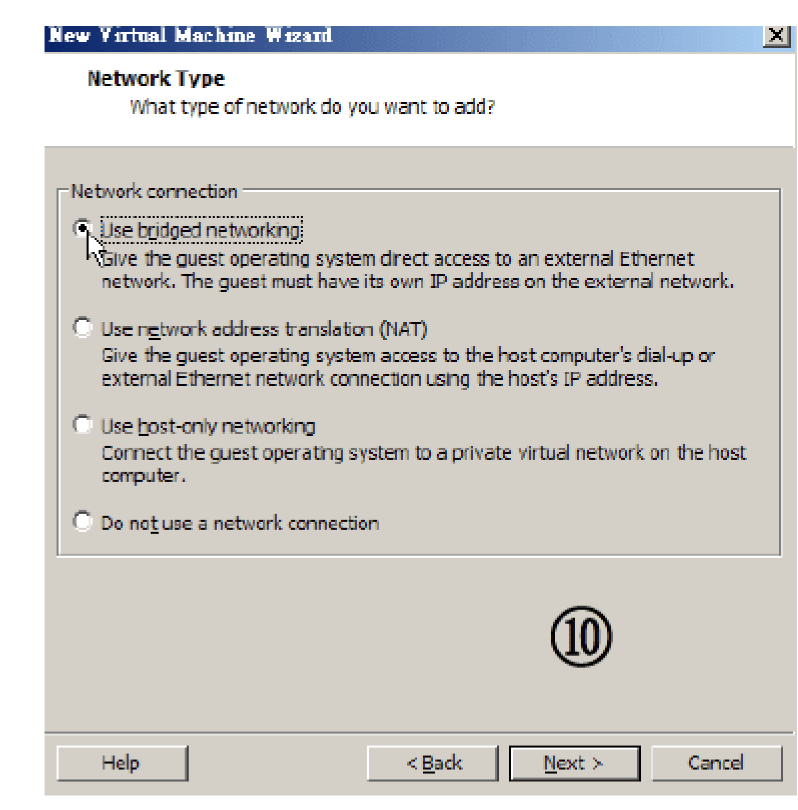



▲ 替这个 VM 命名 ▲ 选择 vCPU

9. 接下来是 VM 的内存，选择 1024MB，单击 Next 按钮继续。

10. 接下来是最重要的地方，即这台 VM 的网卡配置。由于 vCenter 只访问同网段上的 VM，因此只需要一片网卡，选择 Bridged 选项，单击 Next 按钮继续。

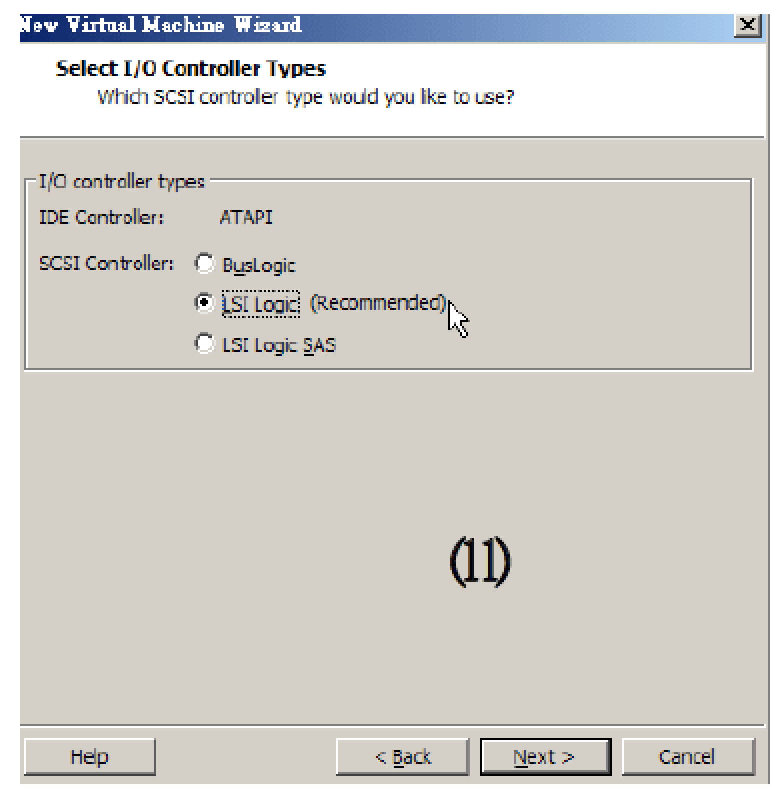


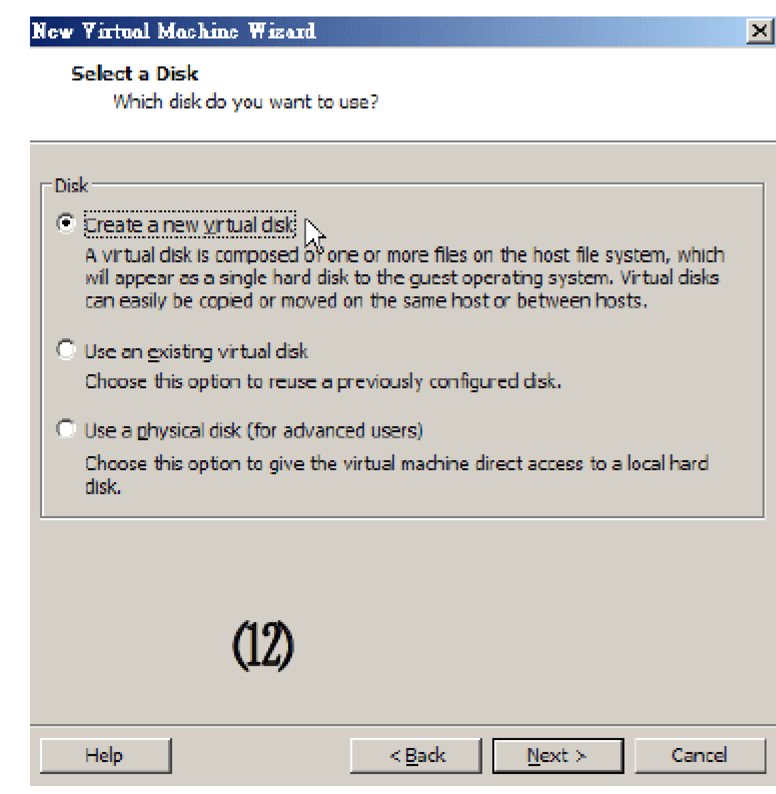


▲ 作为 vCenter，1024MB 就够了 ▲ 这台只连接 Host，因此一片网卡就够了

11. 硬盘选项则使用默认值，单击 Next 按钮继续。

12. 接下来是虚拟硬盘选项，也是使用默认值创建新的虚拟硬盘，单击 Next 按钮继续。

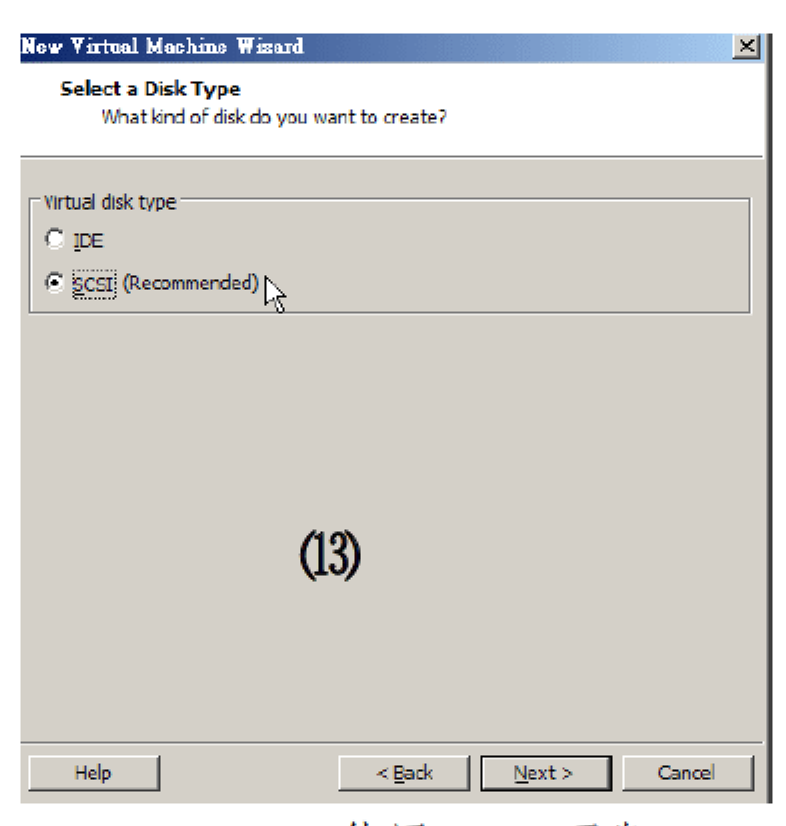




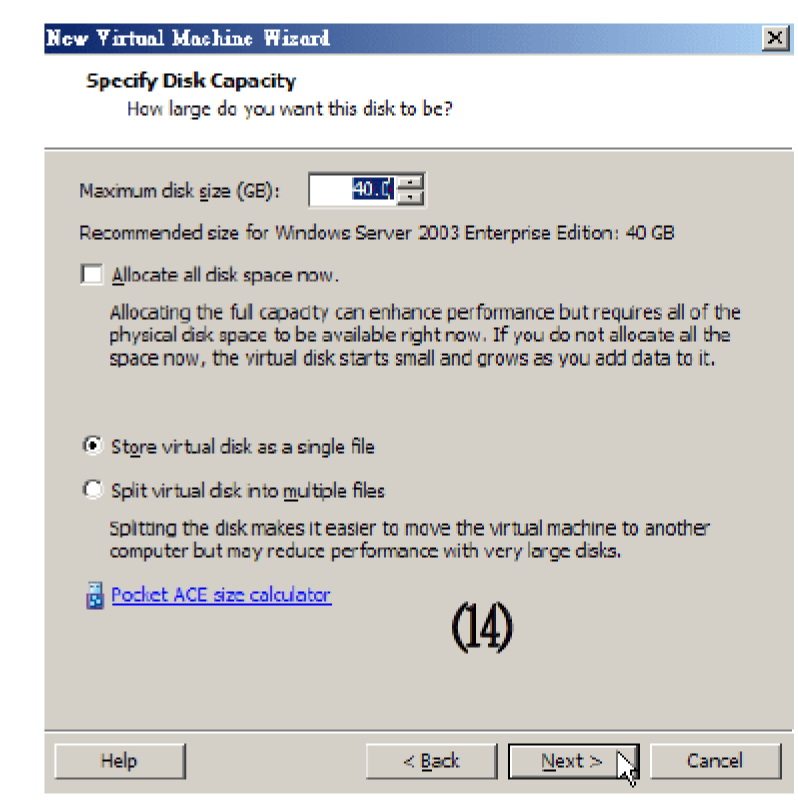
▲ 硬盘使用默认值 ▲ 创建新的硬盘

13. 接下来是硬盘的界面，使用默认值的 SCSI，单击 Next 按钮继续。

14. 接下来是硬盘大小，就使用 40GB 的默认值，单击 Next 按钮继续。



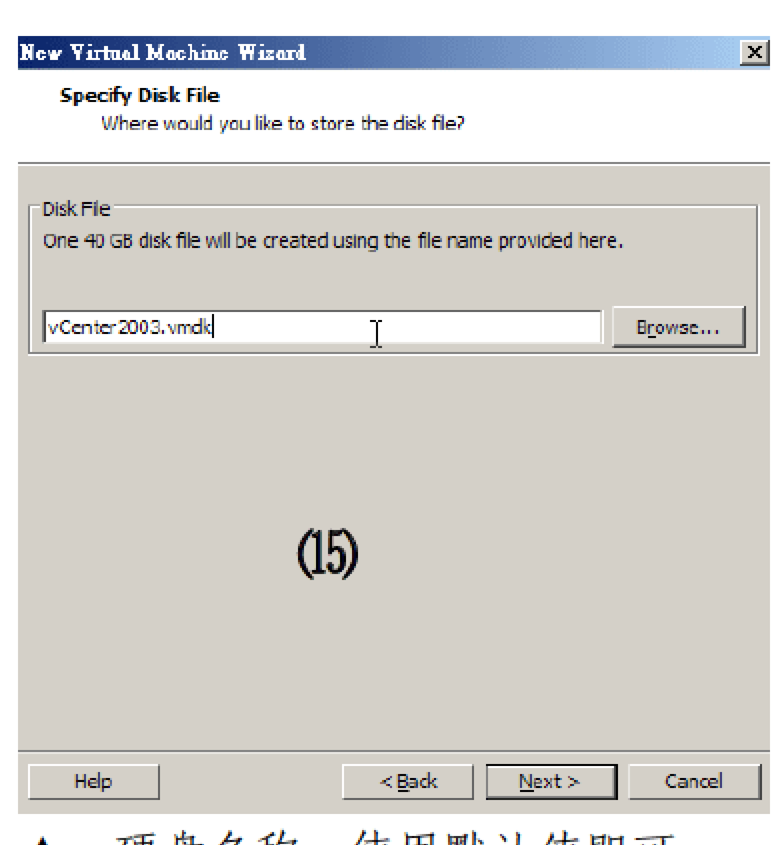
▲ 使用 SCSI 硬盘



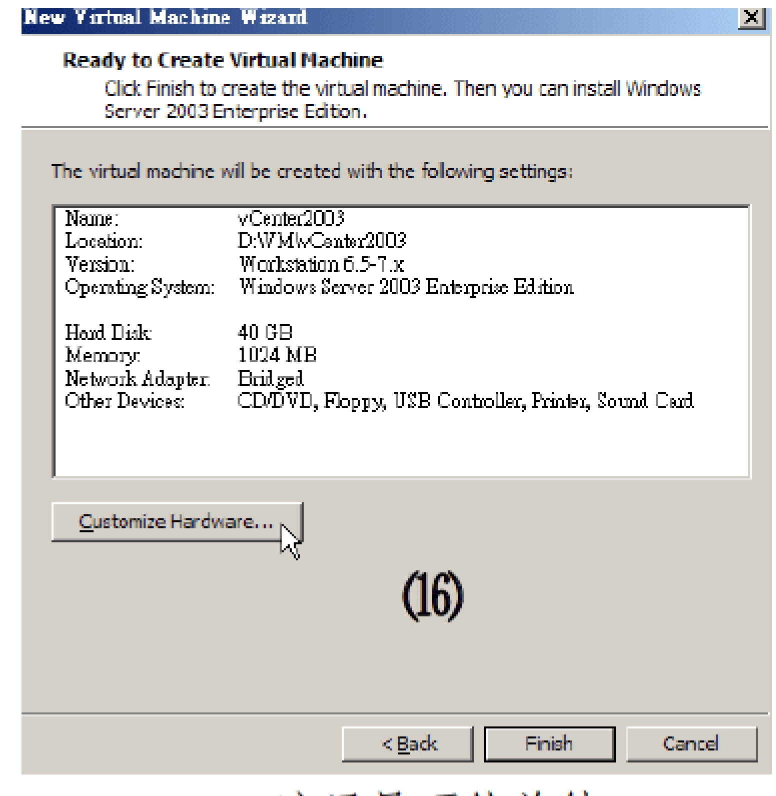
▲ 硬盘大小

15. 接下来是虚拟硬盘的名称，使用默认值，单击 Next 按钮继续。

16. 接下来是硬件的总结，这和 ESX/ESXi 的使用手段大同小异。我们可以单击 Finish 按钮结束配置。

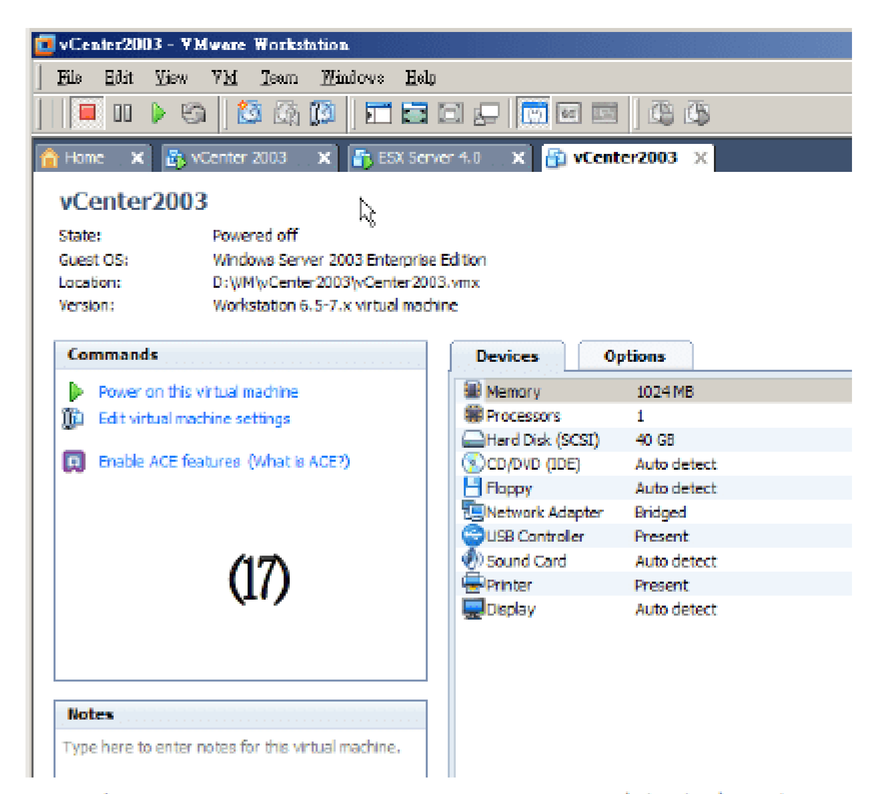


▲ 硬盘名称，使用默认值即可



▲ 这里是硬件总结

17. 可以在 VMware Workstation 的主画面看到已经创建好的 VM 了。



▲ 在 VMware Workstation 下创建好的 VM

2. 安装 Windows 2003 及 vCenter

在 VM 中安装 Windows 2003 与在 ESX/ESXi 中的 VM 安装一模一样，配置好光盘，引导后就可进行安装，这一部分大家可以引用上一章有关 VMware vCenter 的安装详细步骤。

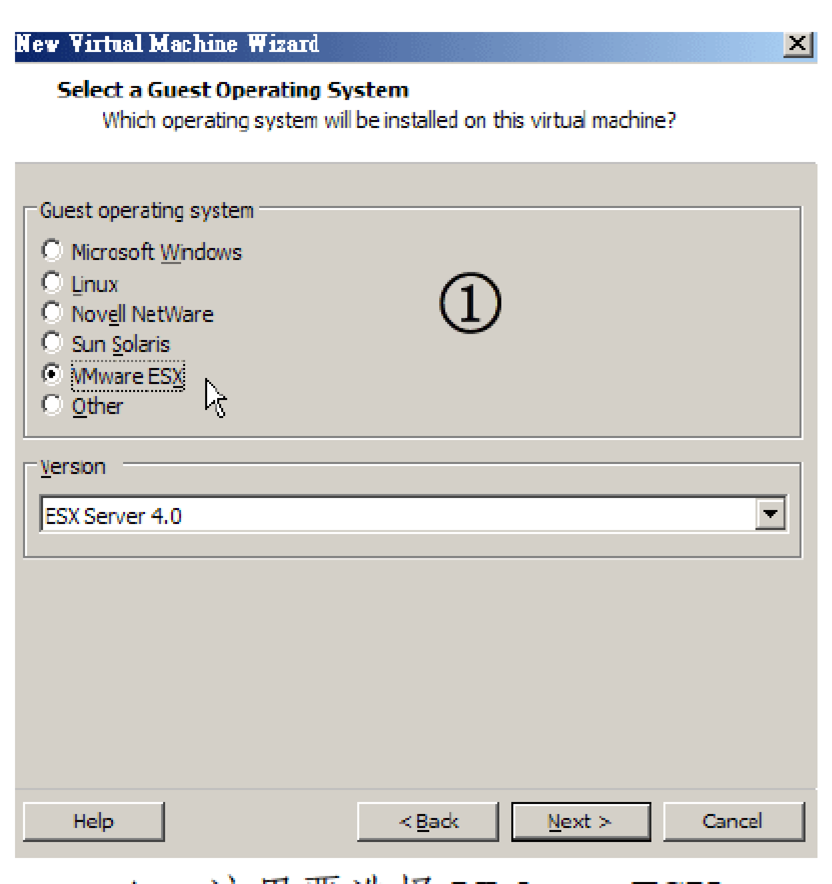
14.2.3　创建及安装 ESX/ESXi 的两台 VM

创建 ESX/ESXi 的 VM 和 Windows 2003 大同小异，VMware Workstation 在 7.0 之后就内置支持在 VM 中安装 ESX/ESXi，并且也可以在 VM 中的 ESX/ESXi 下再安装其他的操作系统，如果是为了实验性质，虽然速度无法满足具体的使用，但功能上已经完全可以使用了。

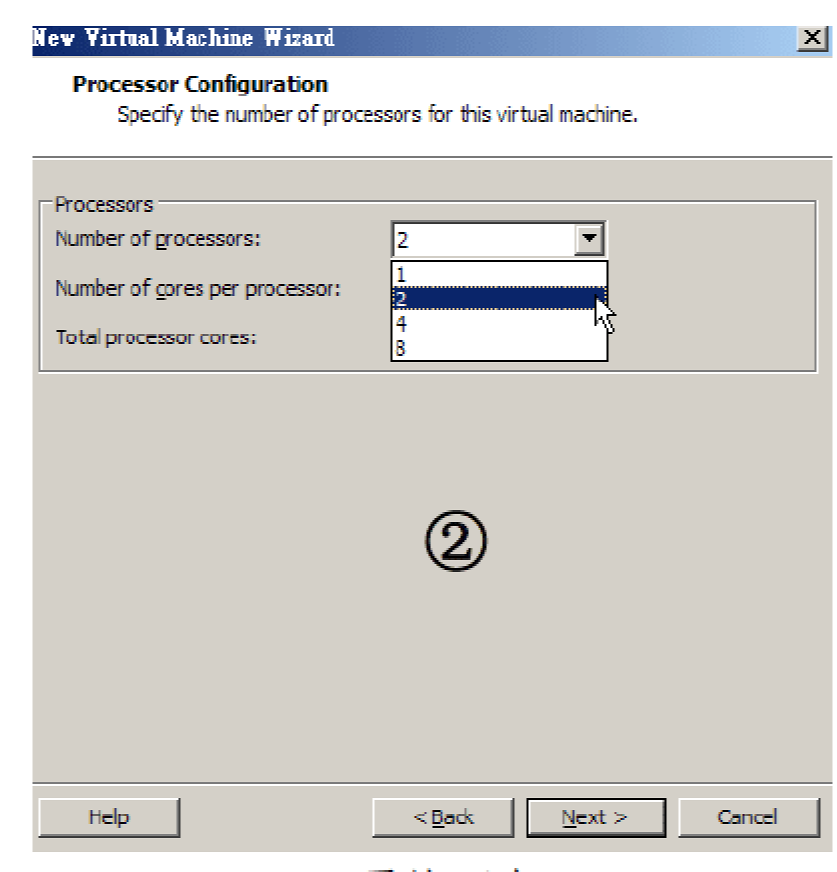
1. 创建 ESX/ESXi 的 VM

在 VMware Workstation 中创建 ESX/ESXi 的 VM 很容易，有几个画面注意一下就行。

1. 在选择客户端操作系统时，选择 VMware ESX，并且在下方的版本处选择 ESX Server 4.0。
2. 在 CPU 选项上，别忘了要给这个 VM 两个 vCPU，速度上才能勉强接受。

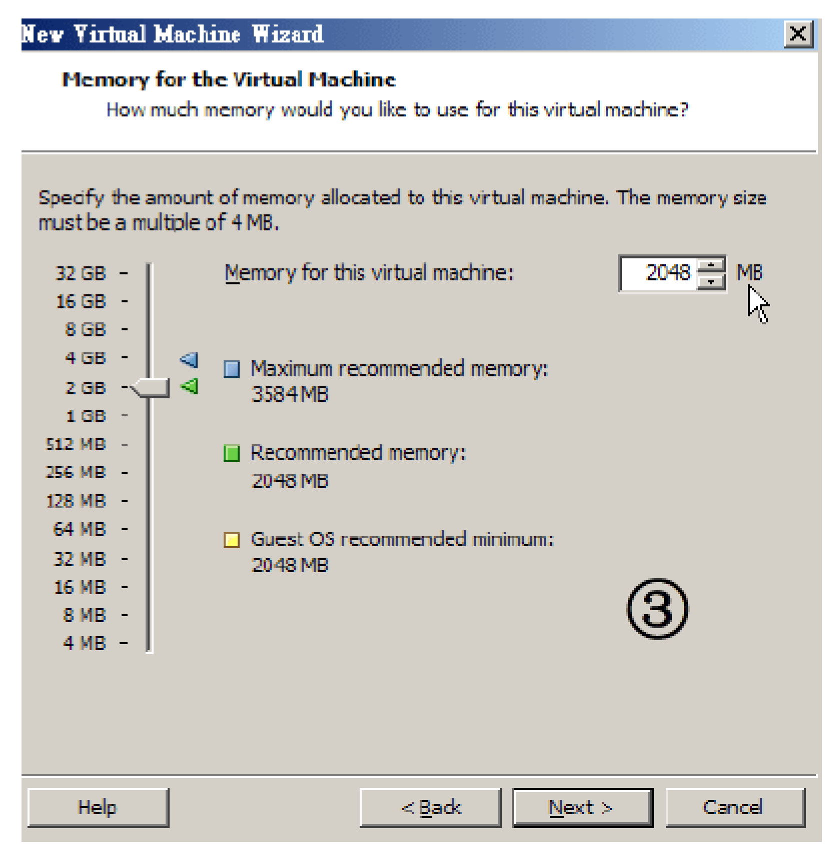


▲　这里要选择 VMware ESX

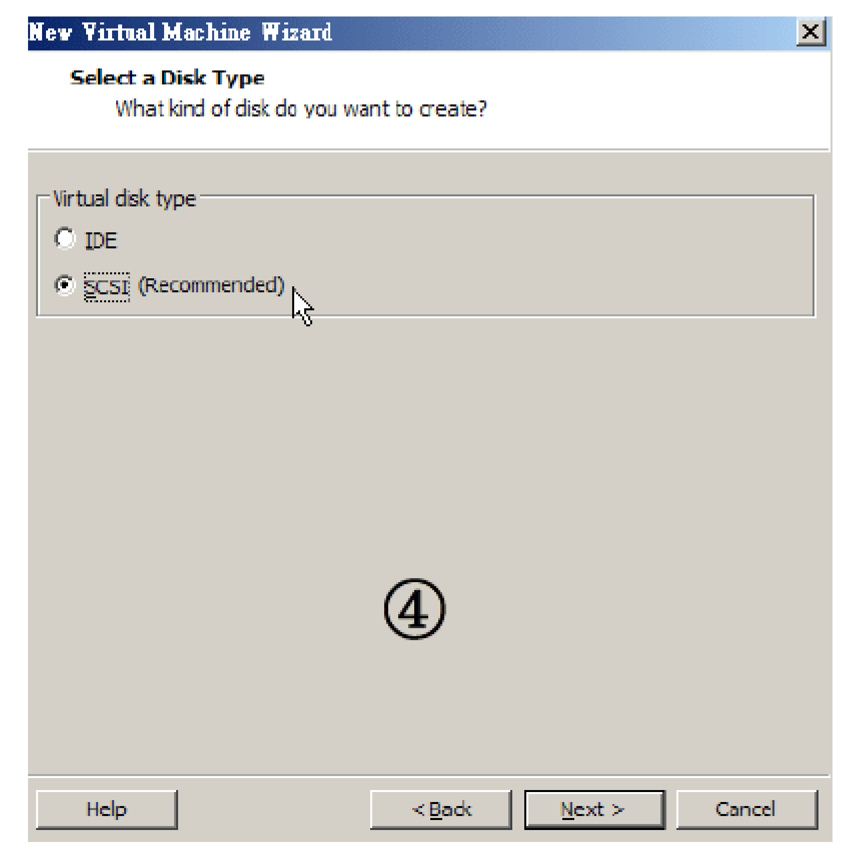


▲　要给两个 vCPU

3. 内存要给到 2048MB。
4. 硬盘要选择 SCSI 界面，ESX/ESXi 不认得 IDE 硬盘。



▲　内存也要给多一点



▲　ESX/ESXi 不认得 IDE 硬盘，要选择 SCSI

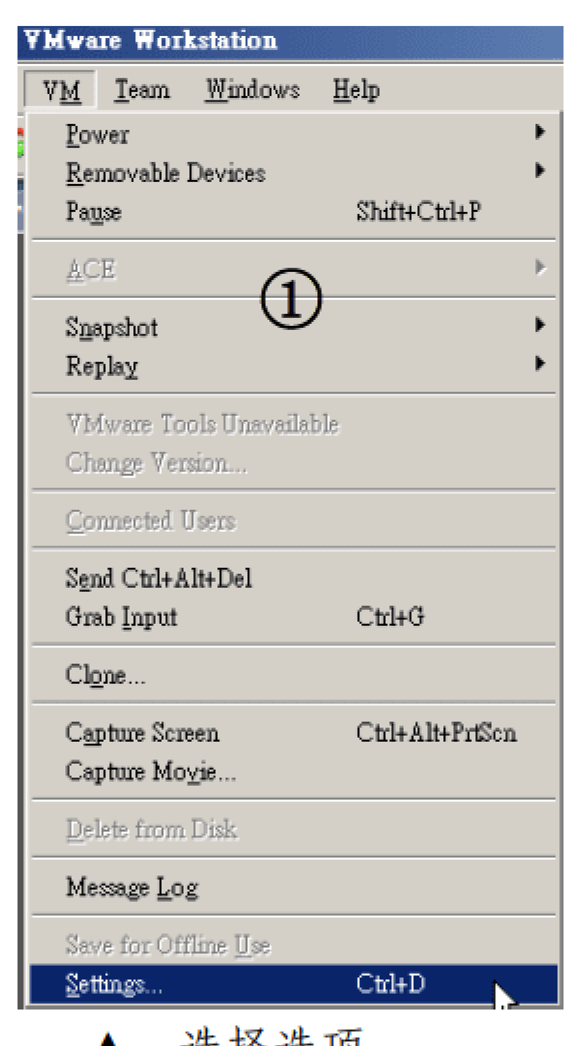
由于需要两台 ESX，因此在创建完一台 ESX 之后，别忘了创建第二台。

2. 添加 ESX VM 的网卡

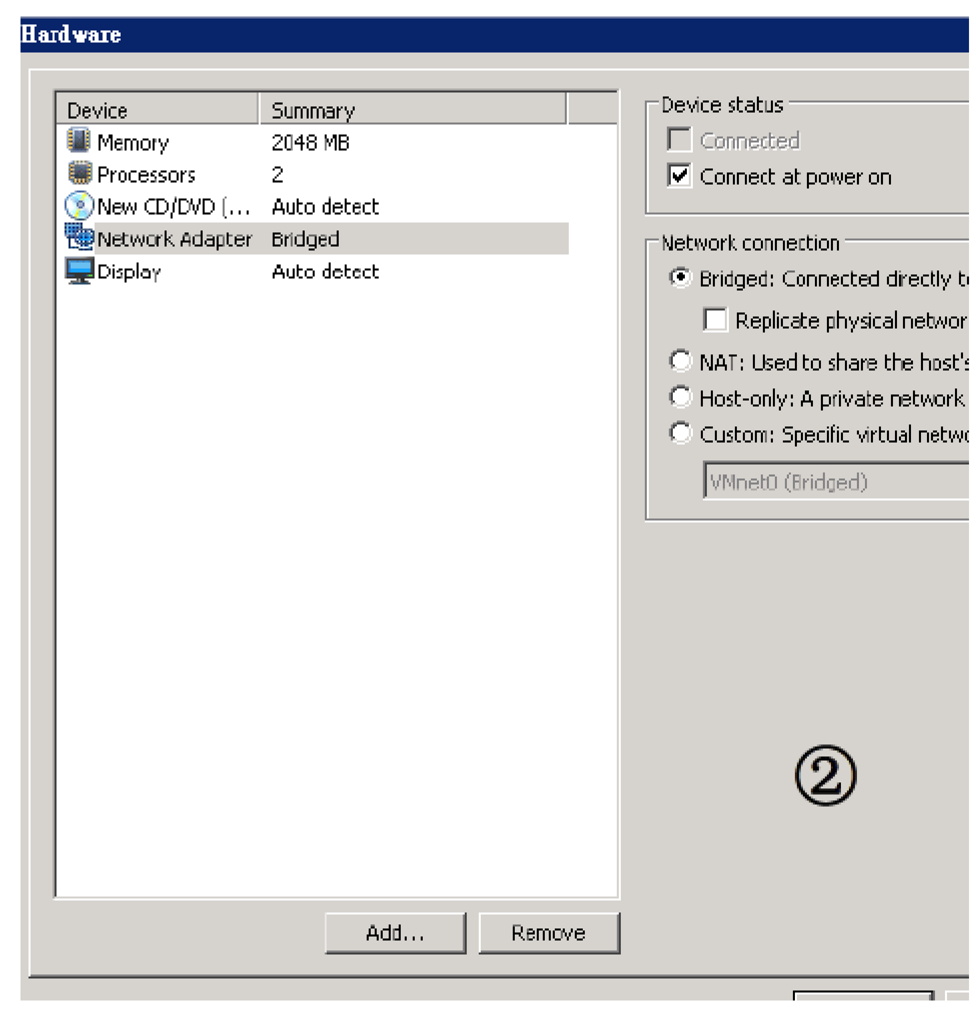
在 VMware Workstation 中的 ESX 需要连上两个网段，因此在创建好 VM 之后，需要再添加网卡，原来在创建 VM 时会有一个网卡，因此我们要再新建一个虚拟网卡。

▶ 在 ESX VM 中添加虚拟网卡

1. 打开刚才创建的 ESX VM，并且选择 VM/Settings 选项。
2. 此时会进入配置的画面，直接单击 Add...按钮。

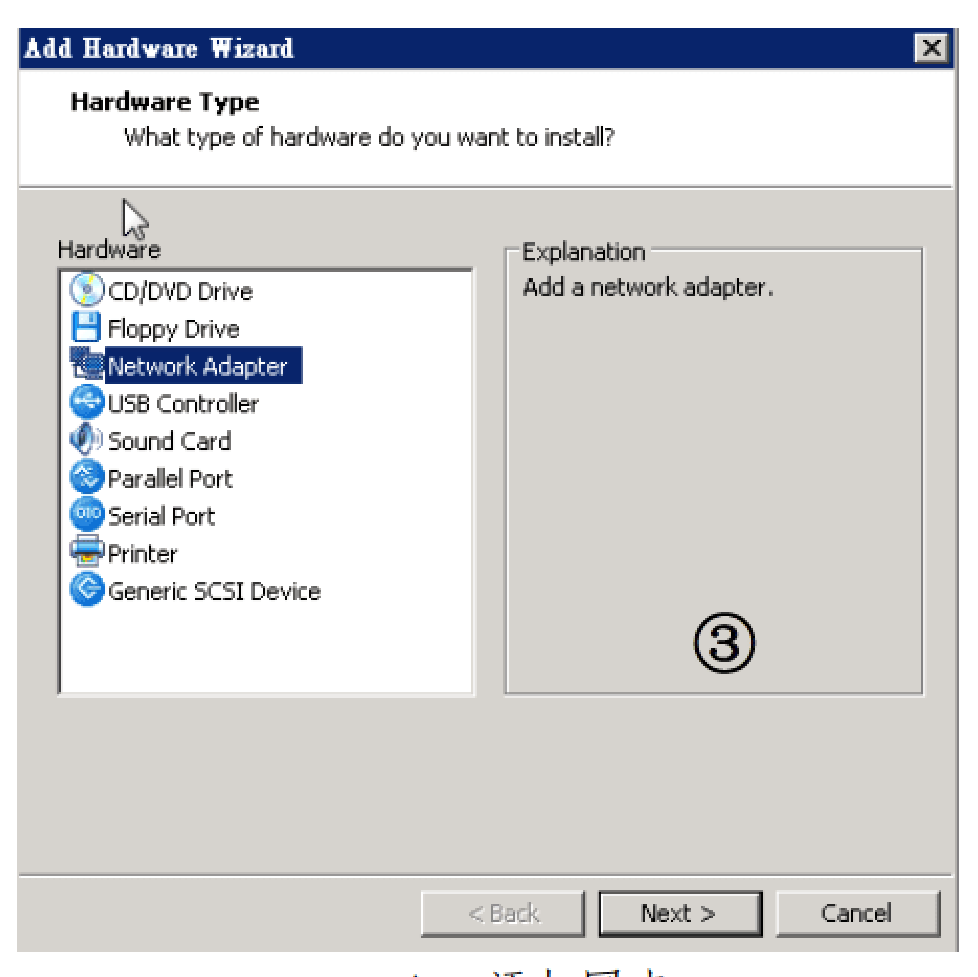


▲ 选择选项

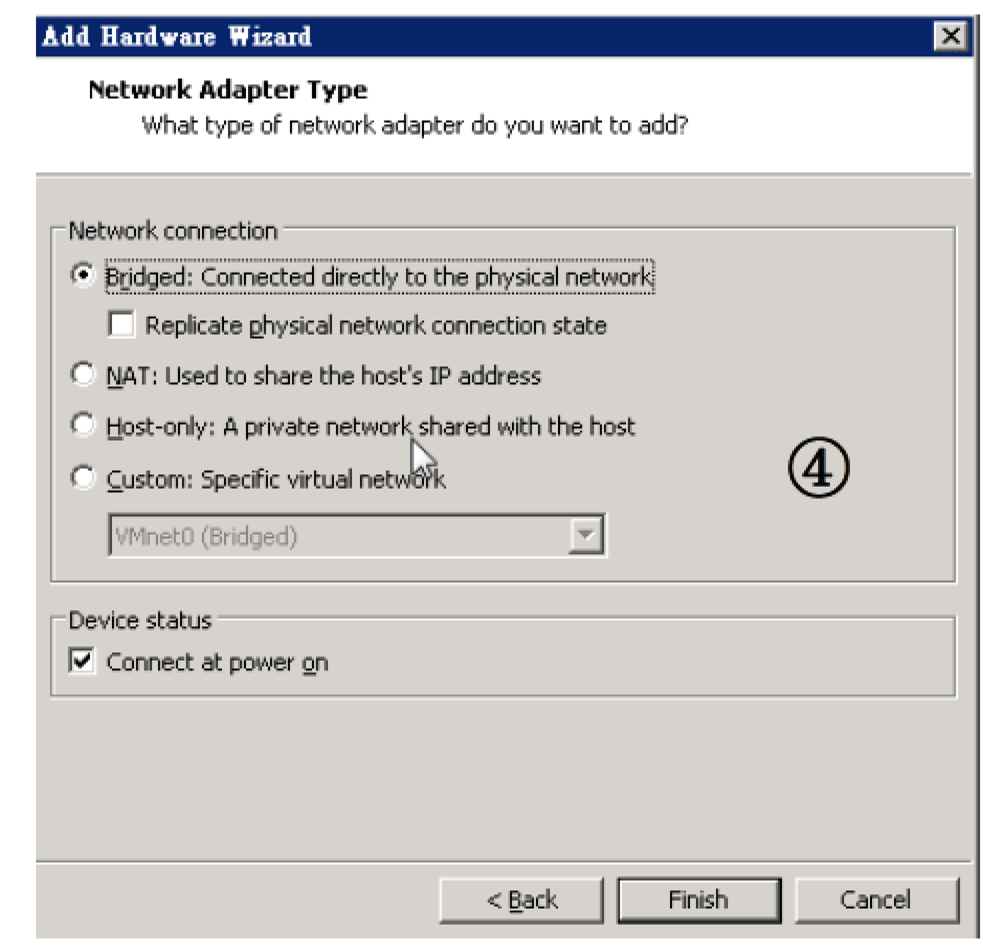


▲ 添加硬件

3. 选择 Network Adapter，并且单击 Next 按钮。
4. 我们先创建一个总控网段的网卡（192.168.1.X）。这个网段的网络桥接是第一片，因此选择第一个选项 Bridged。之后单击 Finish 按钮。

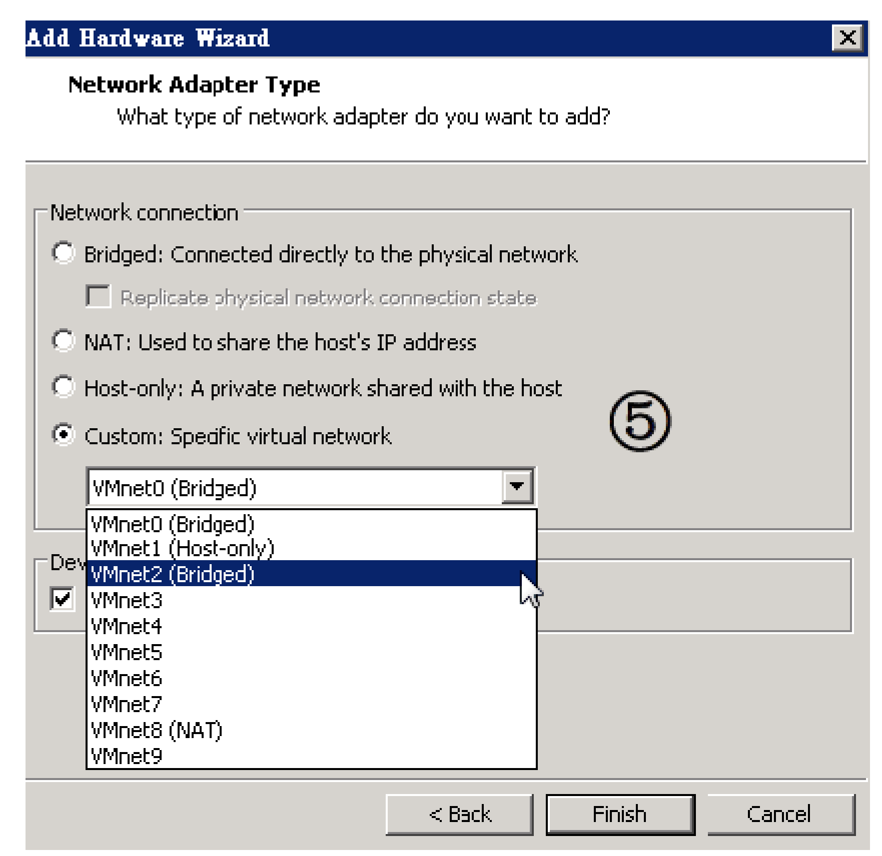


▲ 添加网卡

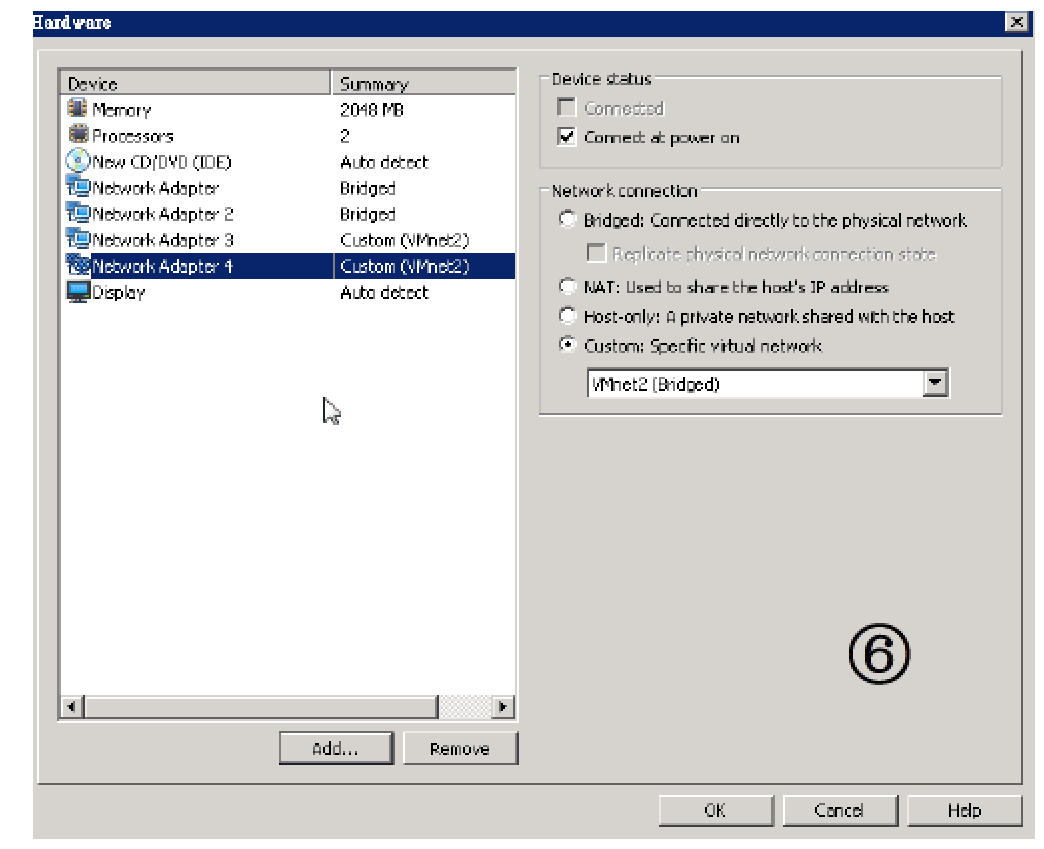


▲ 原先已经有了 192.168.1.X 一张了，再添加一张

5. 接下来会回到硬件配置画面，我们再新建一片网卡，这时候别忘了选择 VMnet2，这是我们配置第二个桥接网段的网卡（10.1.1.X）。
6. 别忘了新建第二片 10.1.1.X 网段的网卡，确定每一台 ESX 上都有四片网卡。



▲ 添加第二张，要选择第二个虚拟路由器

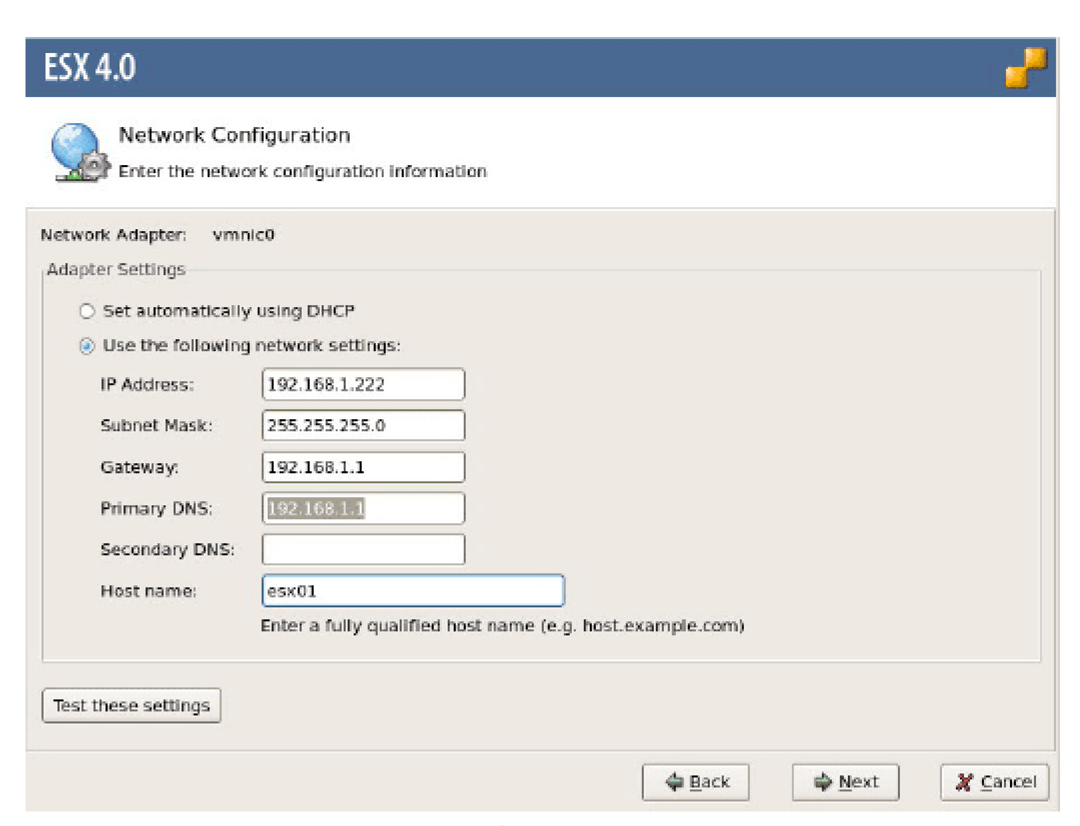


▲ 一台 ESX 要有四片网卡

在第二台 ESX 的 VM 上也创建完全一样的四片虚拟网卡。

3. 安装 ESX

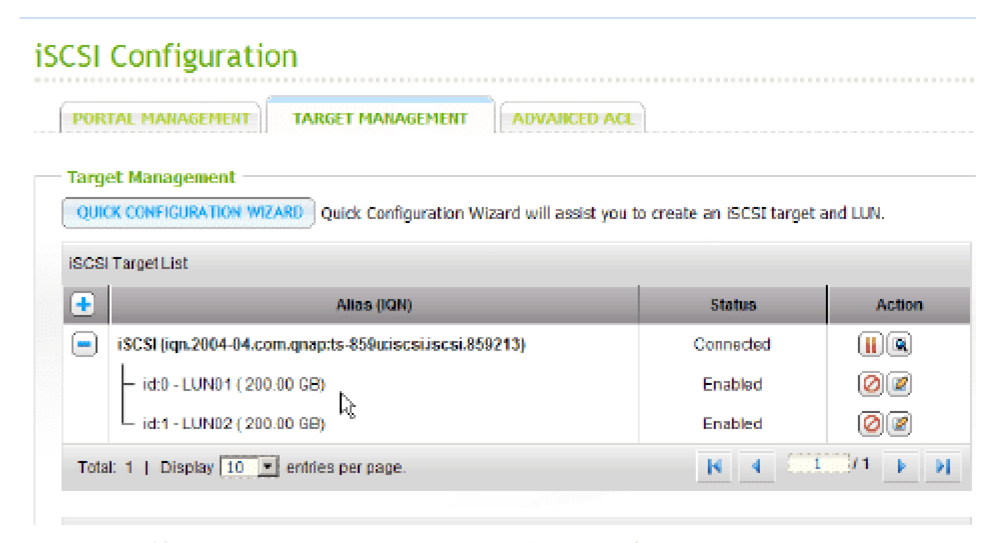
安装 ESX 就如同在物理机上安装一样，但在安装的过程中，别忘了在 IP 及网络配置的地方键入正确的 IP 及主机名，并且确定使用第一片虚拟网卡作为 ESX 默认的 Service Console 总控台，至于其他的网络配置，我们会在需要的地方说明。



▲ 在安装 ESX 时，一定要键入主机名，要不然在 Performance 会无法显示

4. 存储设备的配置

在实验环境的 vSphere 中，我们还是得有独立的存储设备。读者们可以引用第 11 章中 iSCSI 的配置，从一个物理的 NAS 创建 iSCSI 的 LUN。如果读者们没有物理的 NAS 设备，也可以引用第 12 章中 OpenFiler 的 iSCSI 创建 LUN 来使用。在这测试环境中，我们创建两个 LUN，分别是 LUN0 和 LUN1，其大小都是 200G。



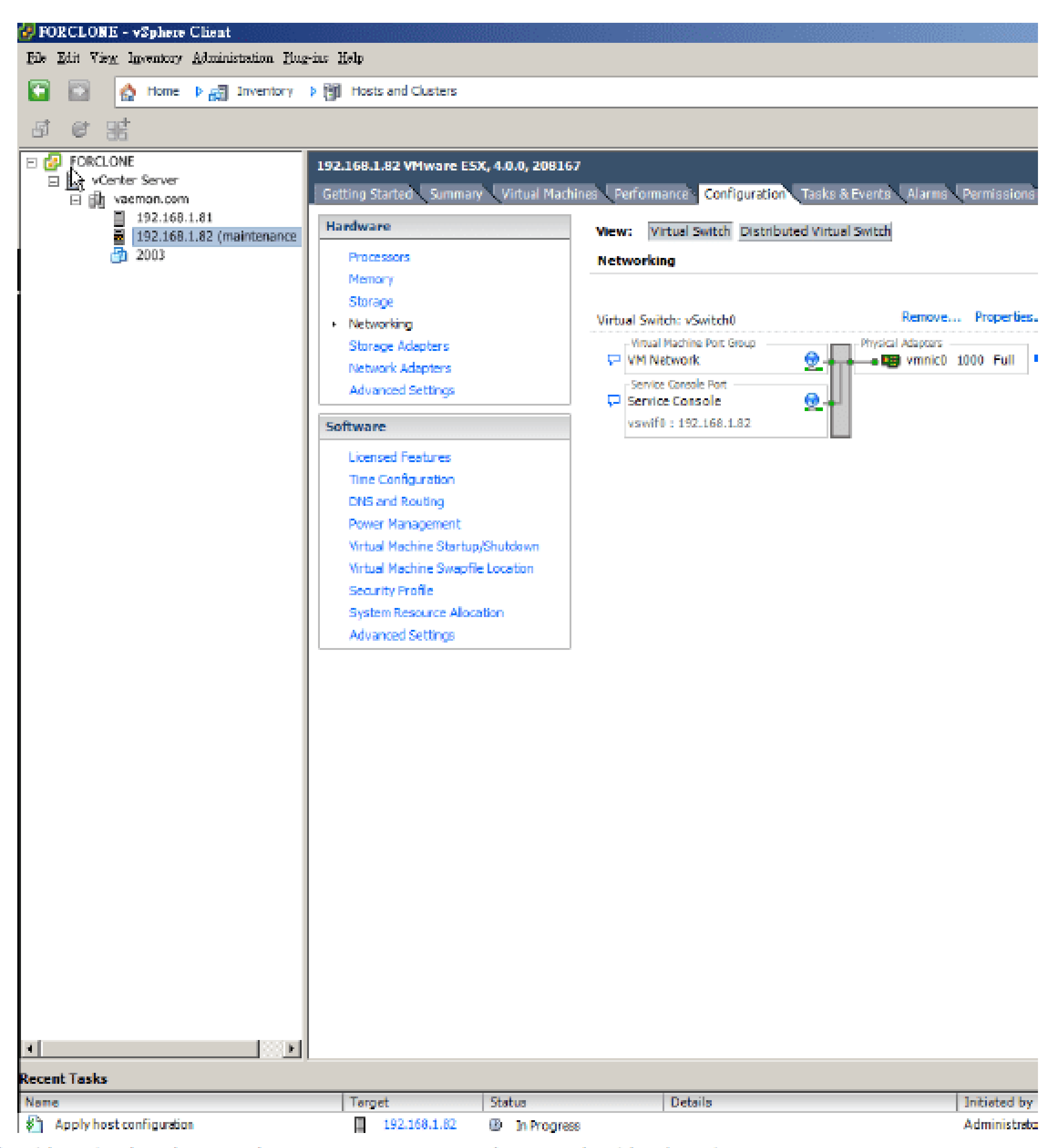
▲ 使用 iSCSI 的硬设备，并且有两个 LUN

14.3 实验环境 vSphere 的网络实践

在 VMware Workstation 中将实验环境架设好之后，我们就必须开始配置网络功能了。由于我们有两个网段，四片网卡，因此我们将使用 NIC Team 的方式，让每一个网段都有两片网卡组成的 vSwitch，以防止某片网卡故障。而每一个 vSwitch 上也都有必须的 Service Console、Virtual Machine Group 以及需要的 VMKernel，如 VMotion、iSCSI 以及 Fault Tolerance Logging 等。

14.3.1 创建双网卡的 vSwitch

在安装 ESX 之后，系统会使用第一片网卡（vmnic0）自动创建一个 vSwitch0。并且在上面创建一个 Service Console 以及虚拟机网络 VM Network。我们接下来会在上面新建三个 VMKernel，分别是 iSCSI、VMotion 以及 FT，并且绑定到另一个网卡上去。



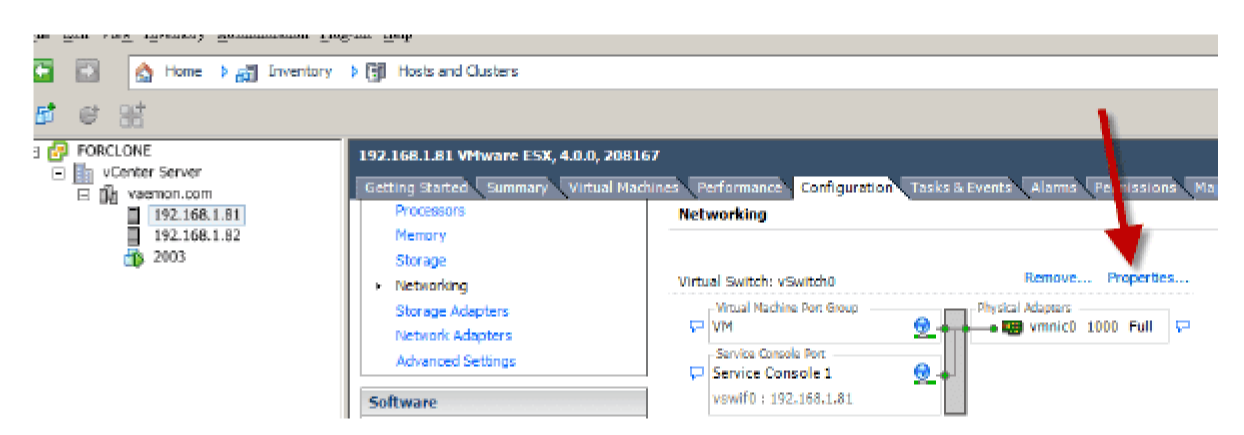
▲ 安装时会有一个 vSwitch，上面有基本的 VM 及 Service Console

1. 创建新的 VMKernel

创建新的 VMKernel 很简单，我们就以创建 VMotion 为例子。

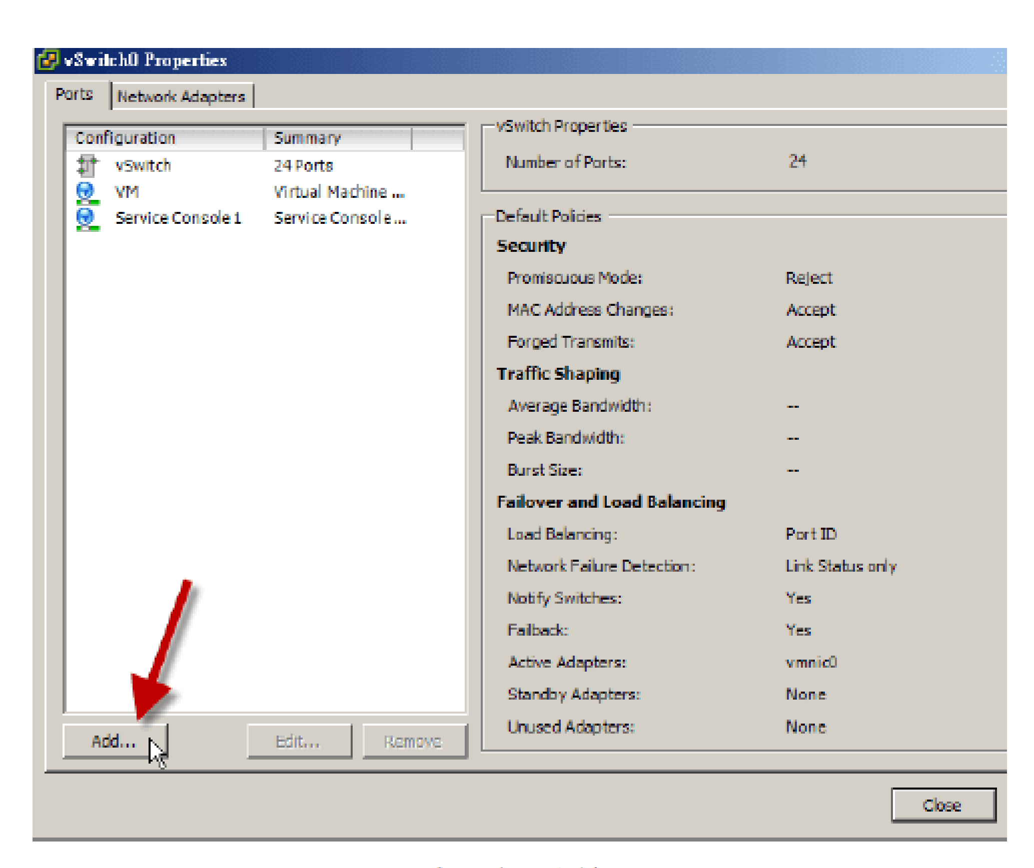
▶ 创建 VMotion 的 VMKernel

1. 进入第一台 ESX，并且选择 Configuration 选项卡，选择 Properties。



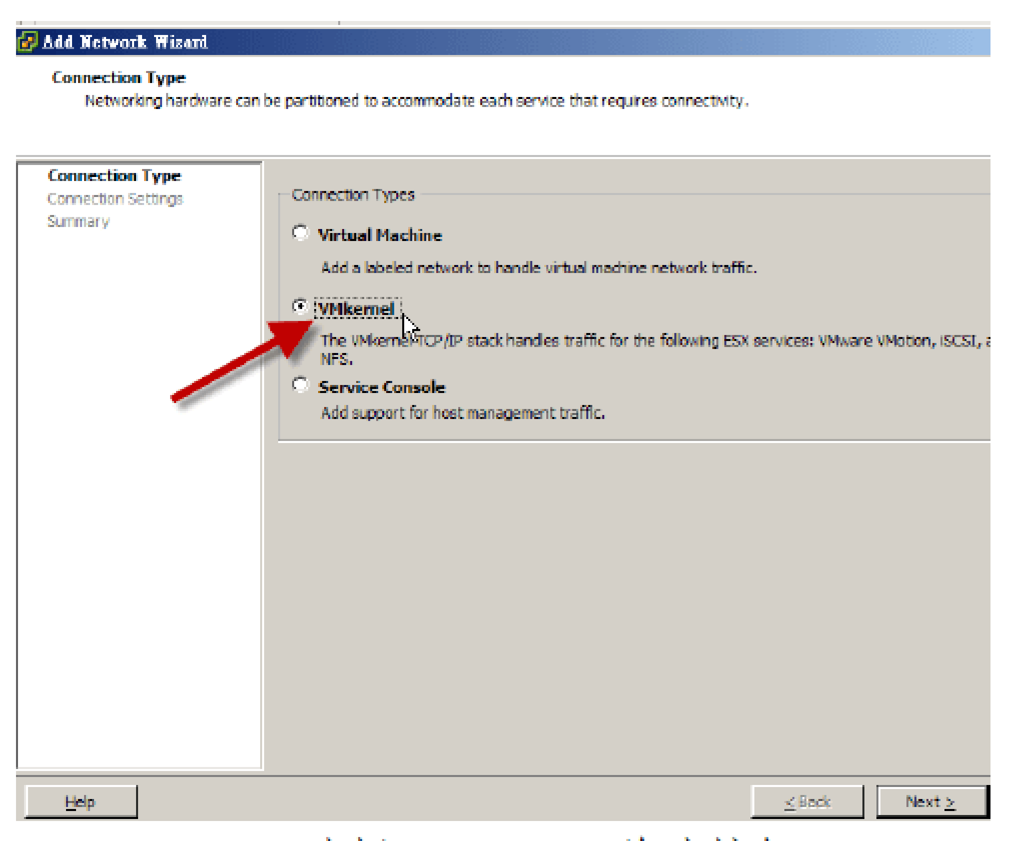
▲ 选择 Properties

2. 单击 Add 按钮。



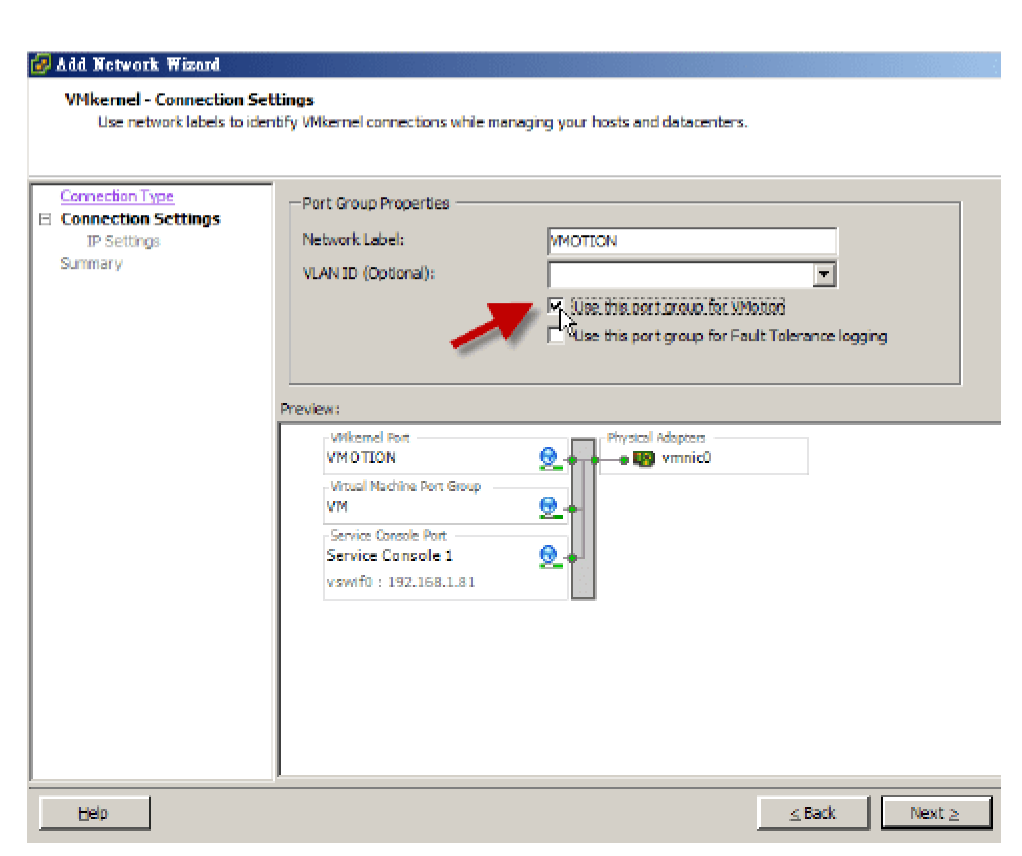
▲ 新建网络 Port

3. 单击 VMkernel 单选按钮，单击 Next 按钮继续。



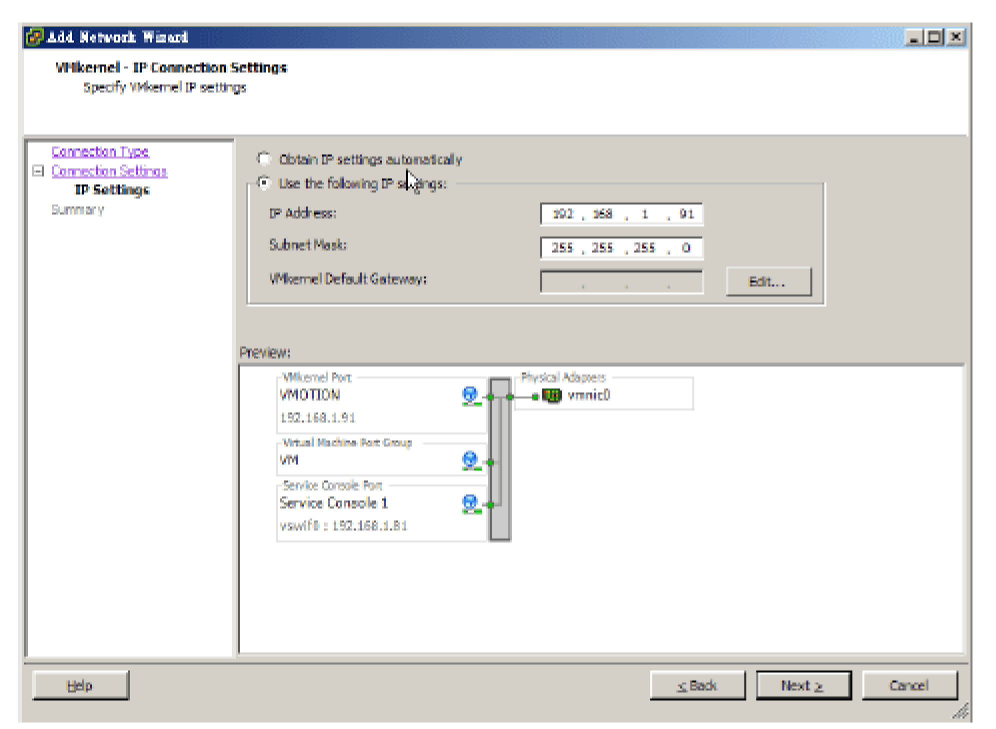
▲ 选择 VMkernel 单选按钮

4. 在 Network Label 列表框键入 VMOTION，注意大小写是有分别的。由于这个 VMkernel 是用来作为 VMotion 的移动，因此我们要把 Use this port group for VMotion 打勾。单击 Next 按钮继续。



▲ 要将这里打勾

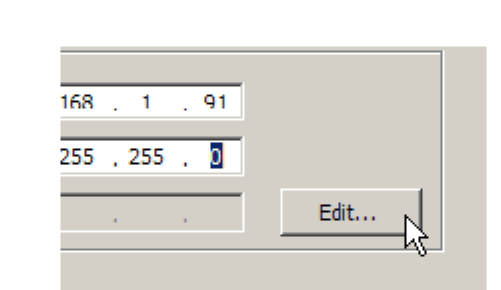
5. 接下来是键入这个 VMOTION 的 IP。键入 192.168.1.91 以及 255.255.255.0。



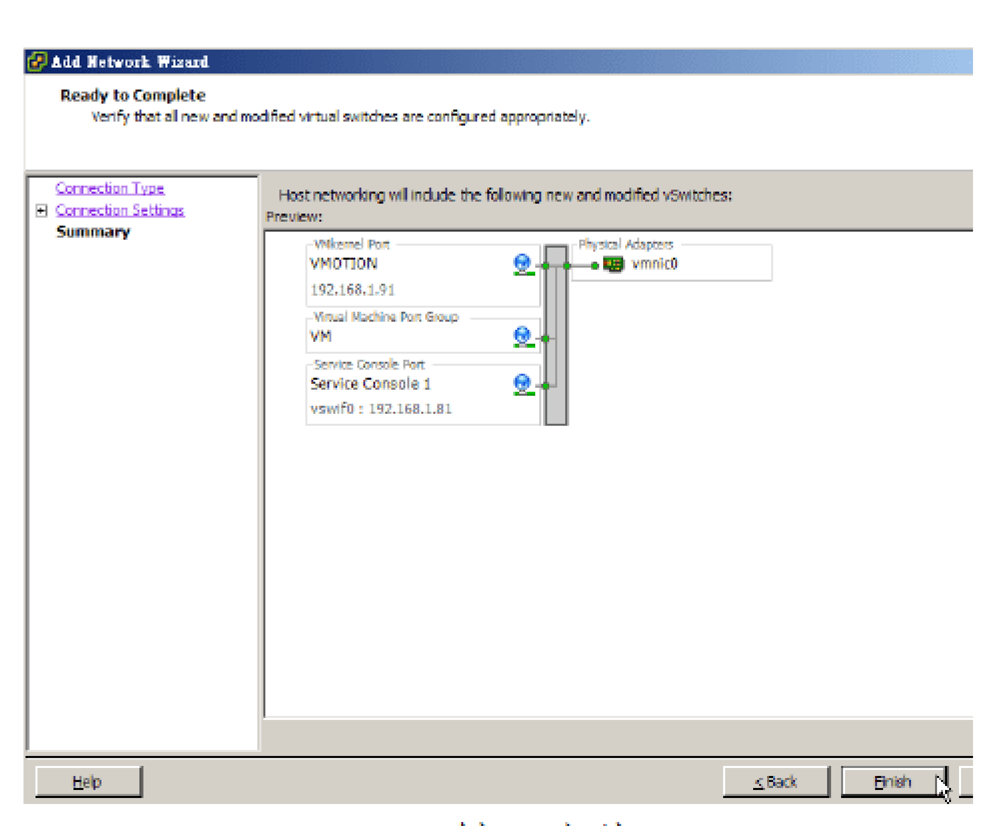
▲ 键入这个 VMkernel 的 IP

6. 接下来单击 Edit 按钮来编辑 Gateway。注意，如果 Gateway 的配置不对，整个 HA 和 VMotion 是无法动作的。

7. 最后是总结画面，如果没有疑义，单击 Finish 按钮落实配置。

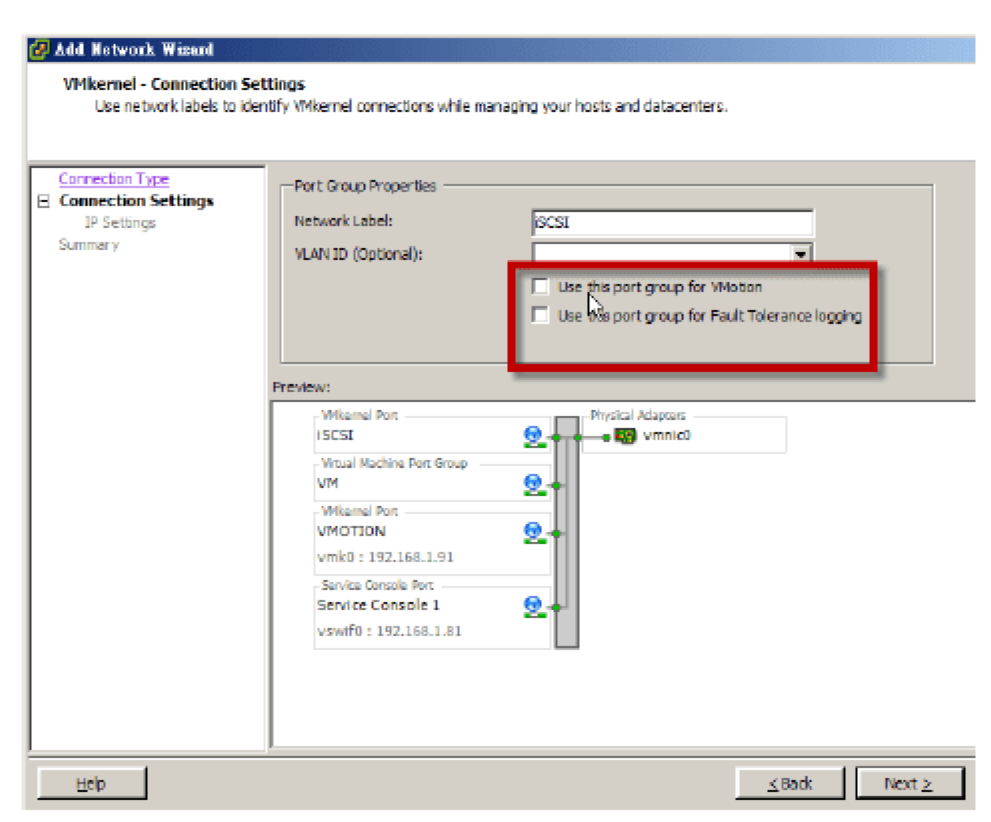


▲ 也要键入 Gateway，要不然 HA 激活不了



▲ 键入完毕

8. 此时我们可以看到 vSwitch0 已经有三个标准的 Port Group。接下来我们再单击 Add 按钮来新建 iSCSI 和 FT。其步骤和新建 VMotion 完全一样。也都是 VMkernel。要注意在新建 iSCSI 时，这两个选项不需要打勾。

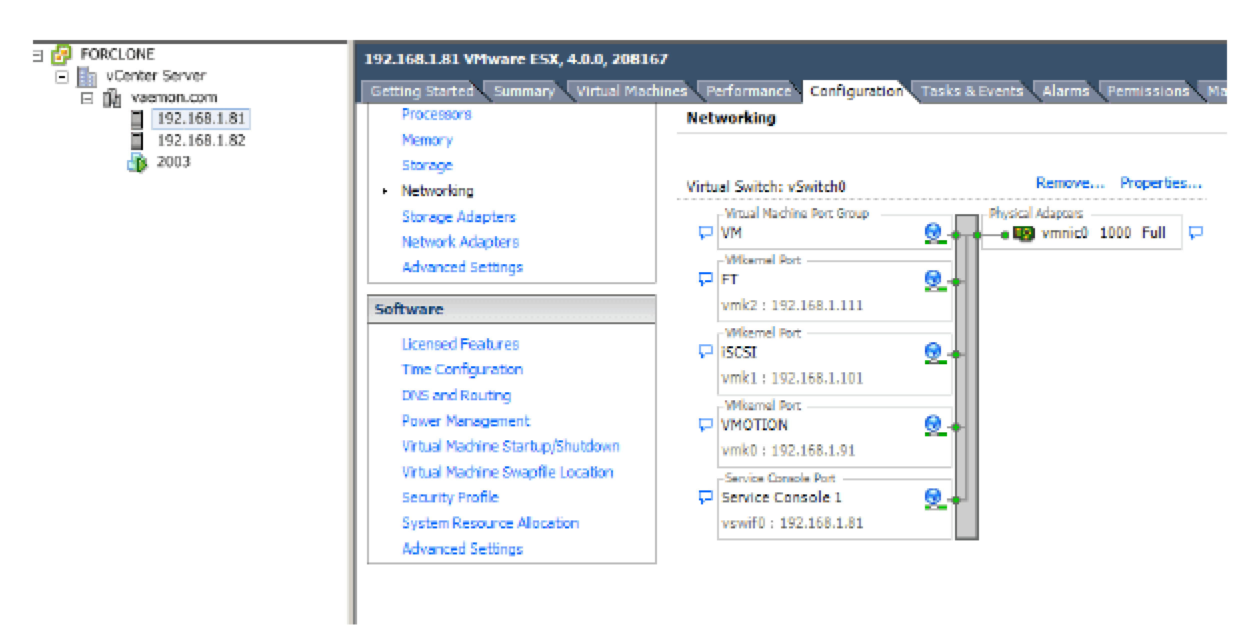


▲ 键入 iSCSI 这里不需要打勾

9. 新建 Fault Tolerance 时，图中选项要打勾。

▲ 要系统支持 FT 功能，这里要打勾

10. 最后 vSwitch0 所有功能都落实了。



▲ 所有的 Port Group 都完成了

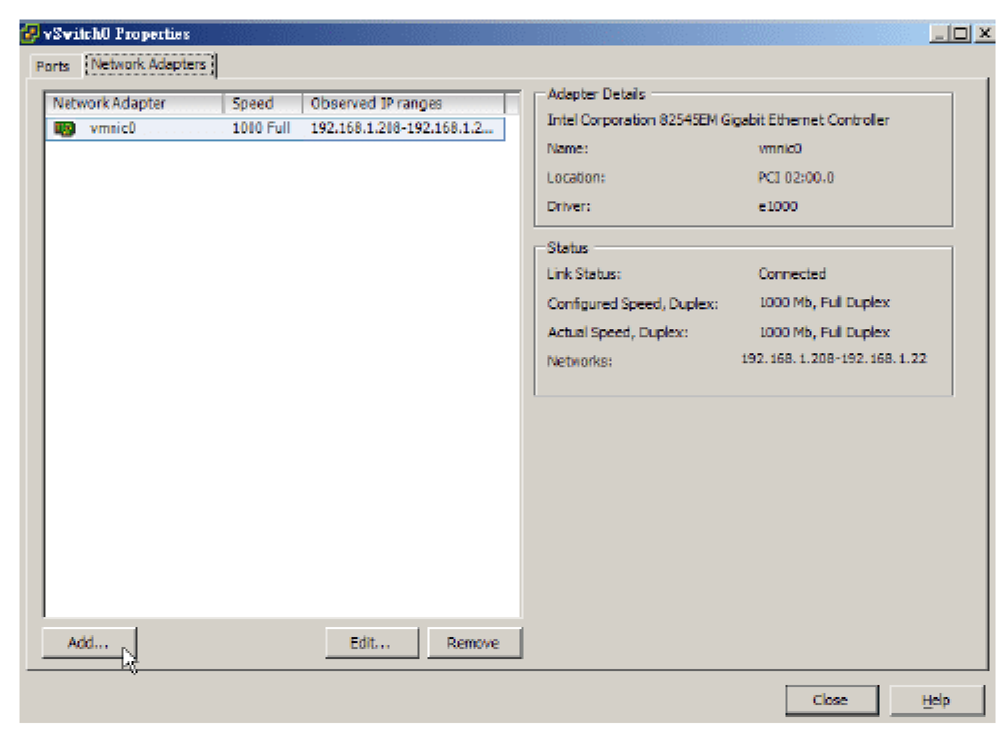
2. 将 vSwtich 使用 NIC Team 绑定第二片网卡

通常为了防止网卡故障，我们会在同一个交换机上插上第二片网卡，使用同一个 IP 网段，并且在 ESX 中将这个冗余的网卡绑上原来的 vSwitch。下面就是详细的步骤。

▶ 将 vSwitch 绑定第二片 NIC Team 网卡

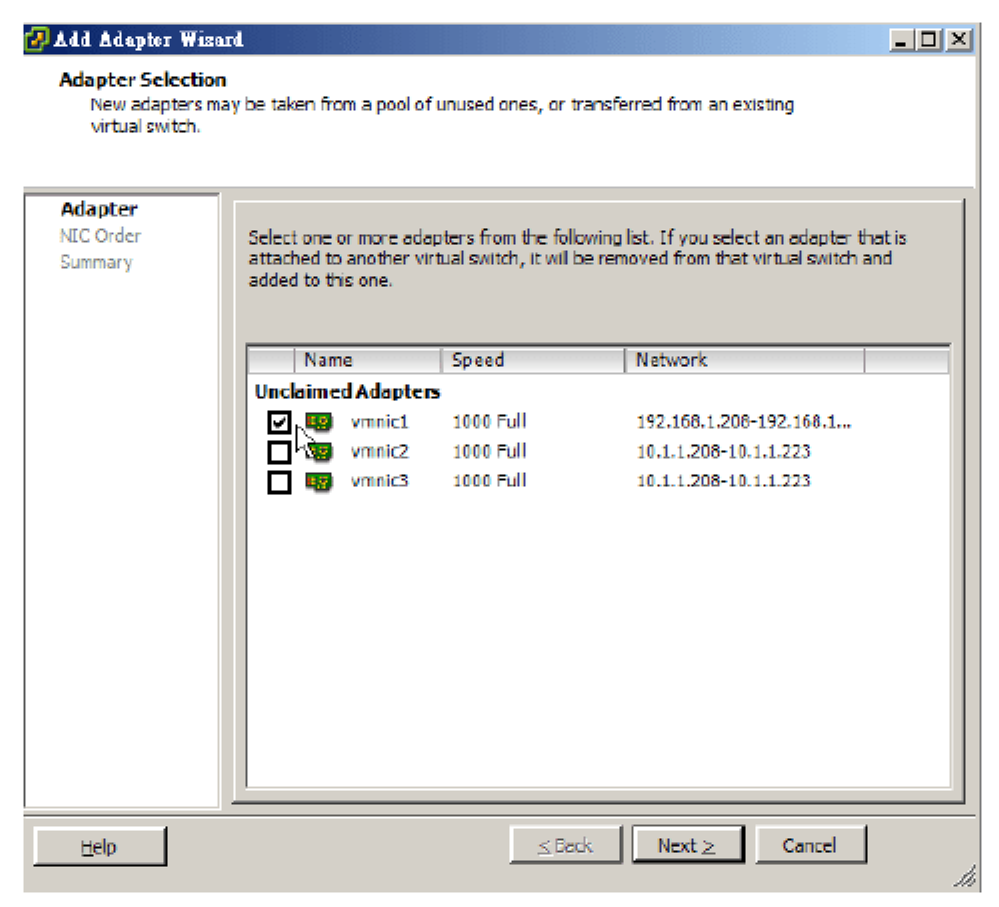
1. 还是选择 Properties。

2. 选择 Network Adapters，并单击 Add 按钮。



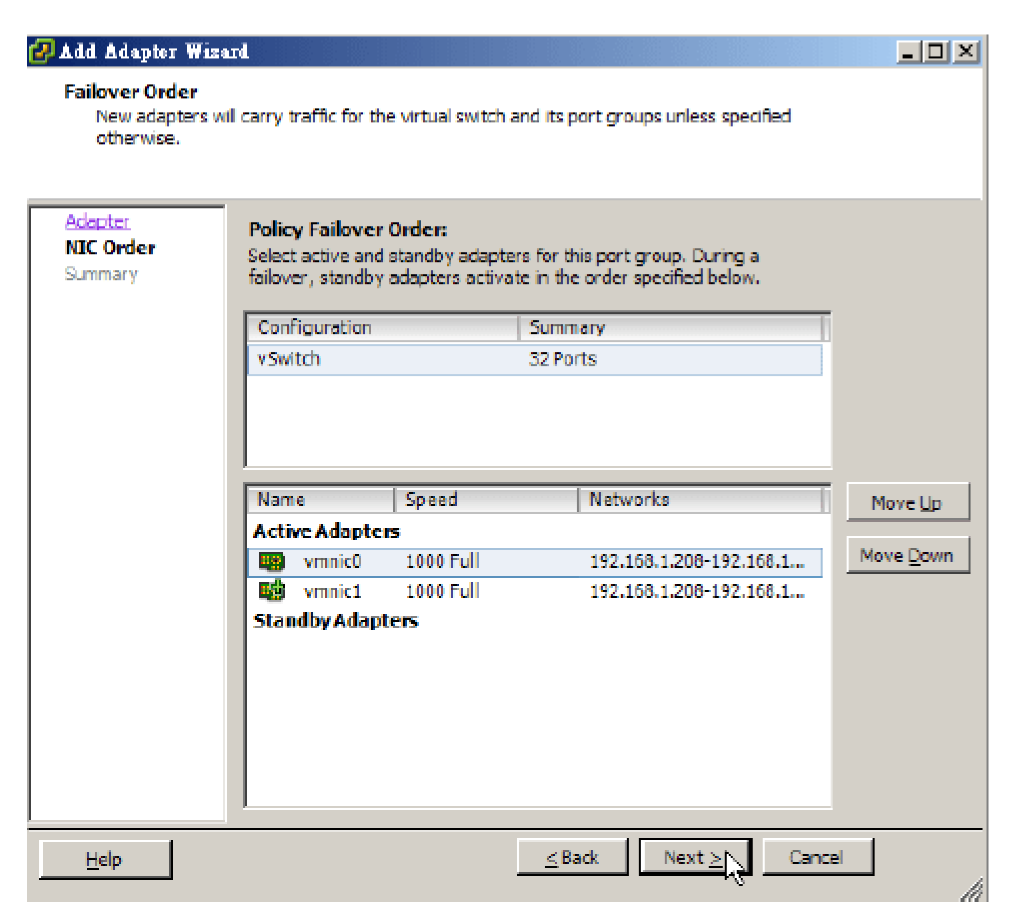
▲ 选择网卡再选择新建

3. 选中同一网段的 vmnic1 并单击 Nextan 按钮。



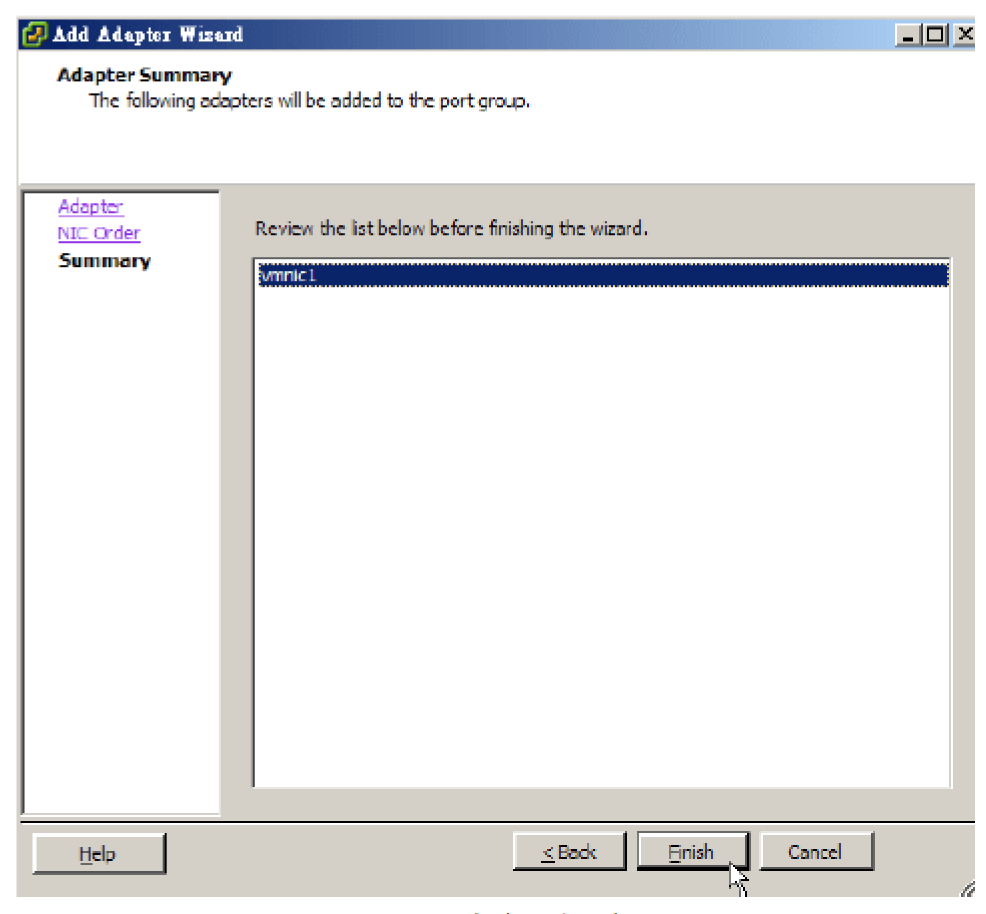
▲ 选取同一网段的网卡

4. 接下来可以调整网卡的顺序，我们就以默认值为主，单击 Next 按钮继续。



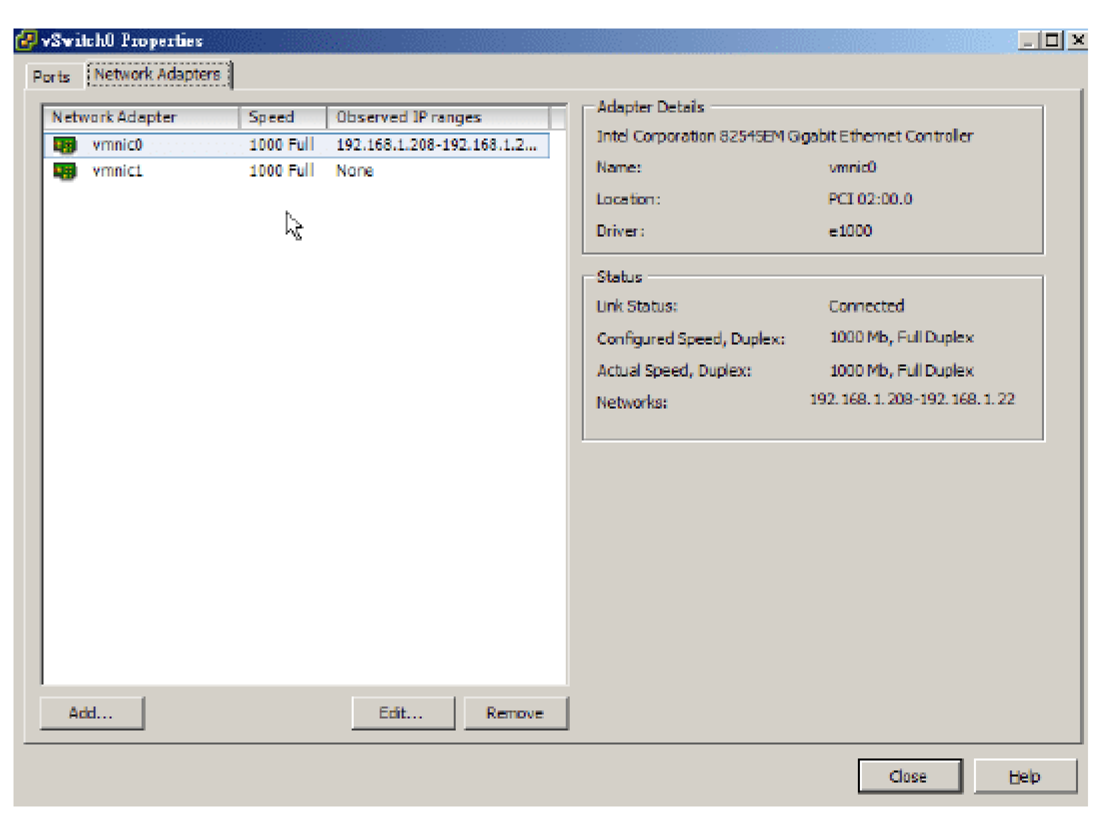
▲ 这里可以选择网卡的顺序，取默认值不选

5. 当弹出如下图所示的画面时，单击 Finish 按钮。



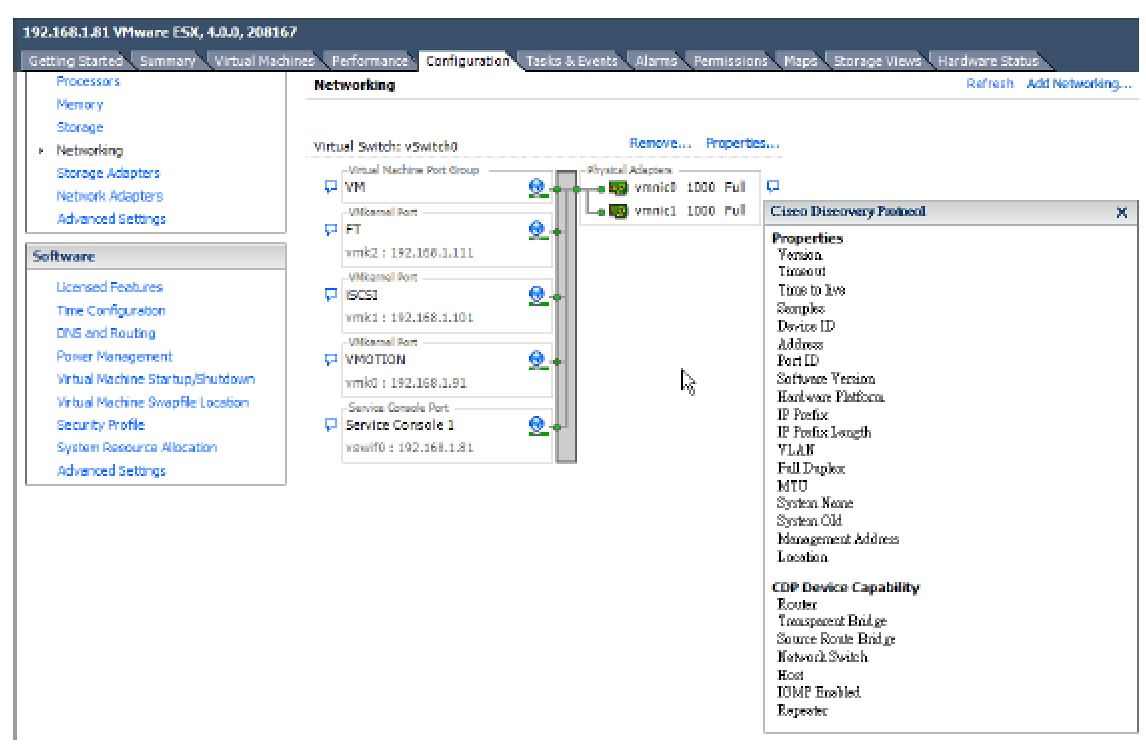
▲ 选择完毕

6. 当弹出如下图所示的画面时，单击 Close 按钮。



▲ 创建落实

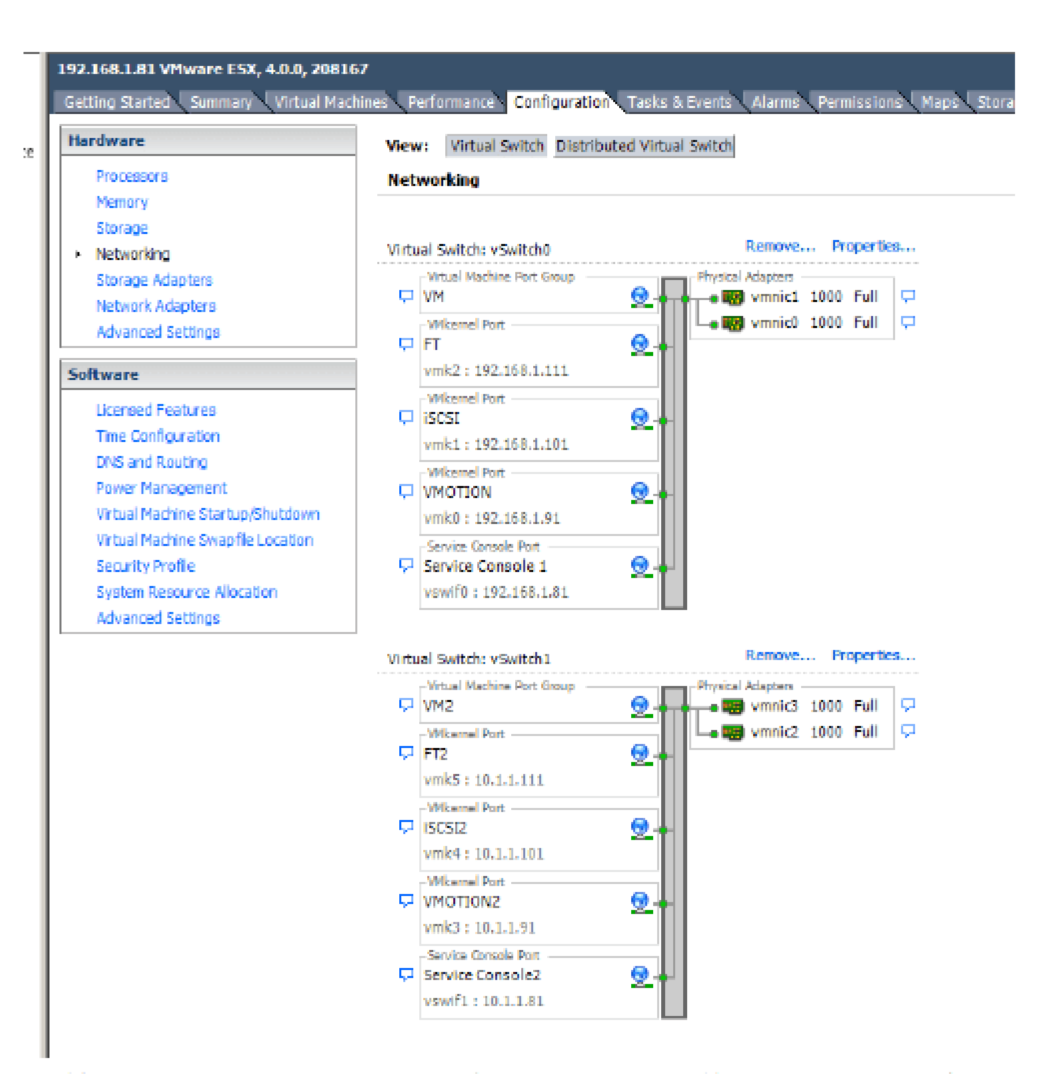
7. 此时可以看到同一个 vSwitch 已经绑定两片网卡了。



▲ 一个 vSwitch 已经有两片网卡了

3. 创建第二个 vSwitch1

当我们创建好第一个 vSwitch0 之后，我们也可以将第三、四片网卡（属于 10.1.1.X 网段），创建第二个 vSwitch1。创建第二个 vSwitch1 的方法可以和 vSwitch1 的配置完全一样，只是所有的 port group 都在后面加一个 2，IP 也以 10.1.1.X 为主，创建好之后可以在网络配置中看到。



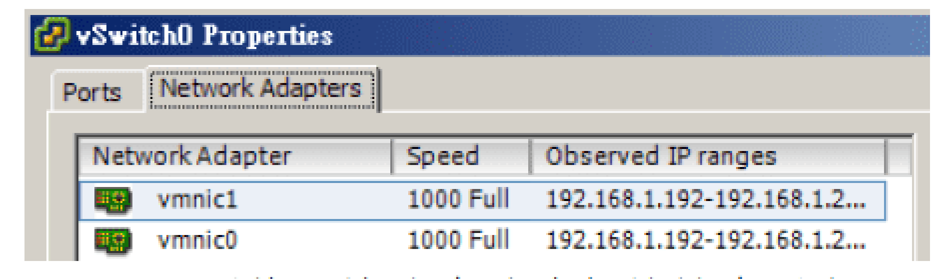
▲ 创建第二个 vSwitch，本章稍后会有使用 Profile 创建的方法

14.4 创建 vSphere 生产环境

绝大部分的 vSphere 都运行在生产环境中，这意味着物理服务器以及独立的存储设备。在生产环境中，一旦实施下去就很难更改原始的设计架构，因此设计一个完整的 vSphere 生产环境比使用上还要重要千百倍。

14.4.1 生产环境的设计原则

一般来说，生产环境最重要的就是防止单点失误，在设计上最重要的就是将所有部件都准备一个副本。另外就是将不同的功能分离。以 vSphere 来说，最重的就是设计网络和存储的多路径，网卡就是多网卡的联机，而存储就是 Multipath。



▲ 网络环境的多重路径就是多网卡

1. 生产环境的服务器

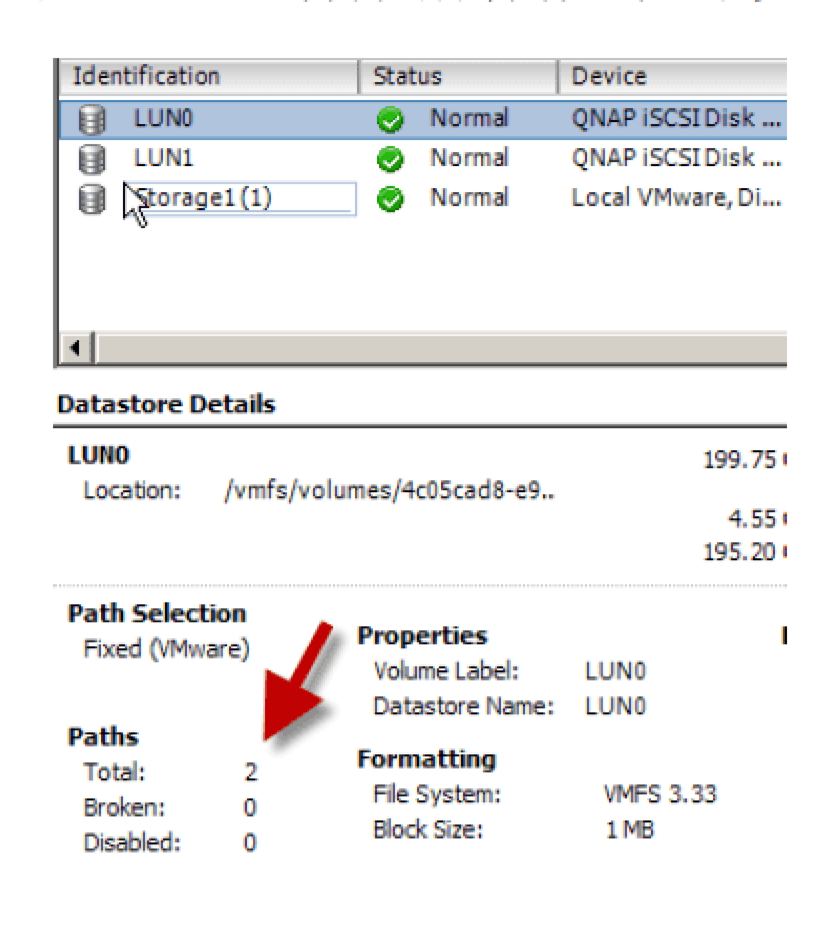
在本书的生产环境中，我们的设计如表 1：

表 1 生产环境的硬设备

	vCenter 主机	ESX Host 1	ESX Host 2
主机	IBM X3550 M2	Dell R721	HP DL380 G6
CPU	Intel Xeon 5500×2	Intel Xeon 5500×2	Intel Xeon 5500×2
内存	8G	8G	8G
硬盘	RAID 1	RAID 1	RAID 1
网卡	GB 网卡×4	GB 网卡×4	GB 网卡×4
第一片网卡	172.17.23.203	172.17.23.29	172.17.23.30
第二片网卡	192.168.1.X	192.168.1.X	192.168.1.X
第三片网卡	192.168.1.X	192.168.1.X	192.168.1.X
第四片网卡	172.17.23.X	172.17.23.X	172.17.23.X

2. 生产环境的存储设备

本书的生产环境存储设备使用 iSCSI 物理 NAS。使用的是 QNAP 的 TS-859 Pro Turbo， GB 网卡，网络 IP 为 192.168.1.24 及 192.168.1.27 两个。在 ESX 主机上将使用 192.168.1.X 的网段连接存储设备。我们会在下一章的 VMotion 部分具体操作生产环境的网络配置。



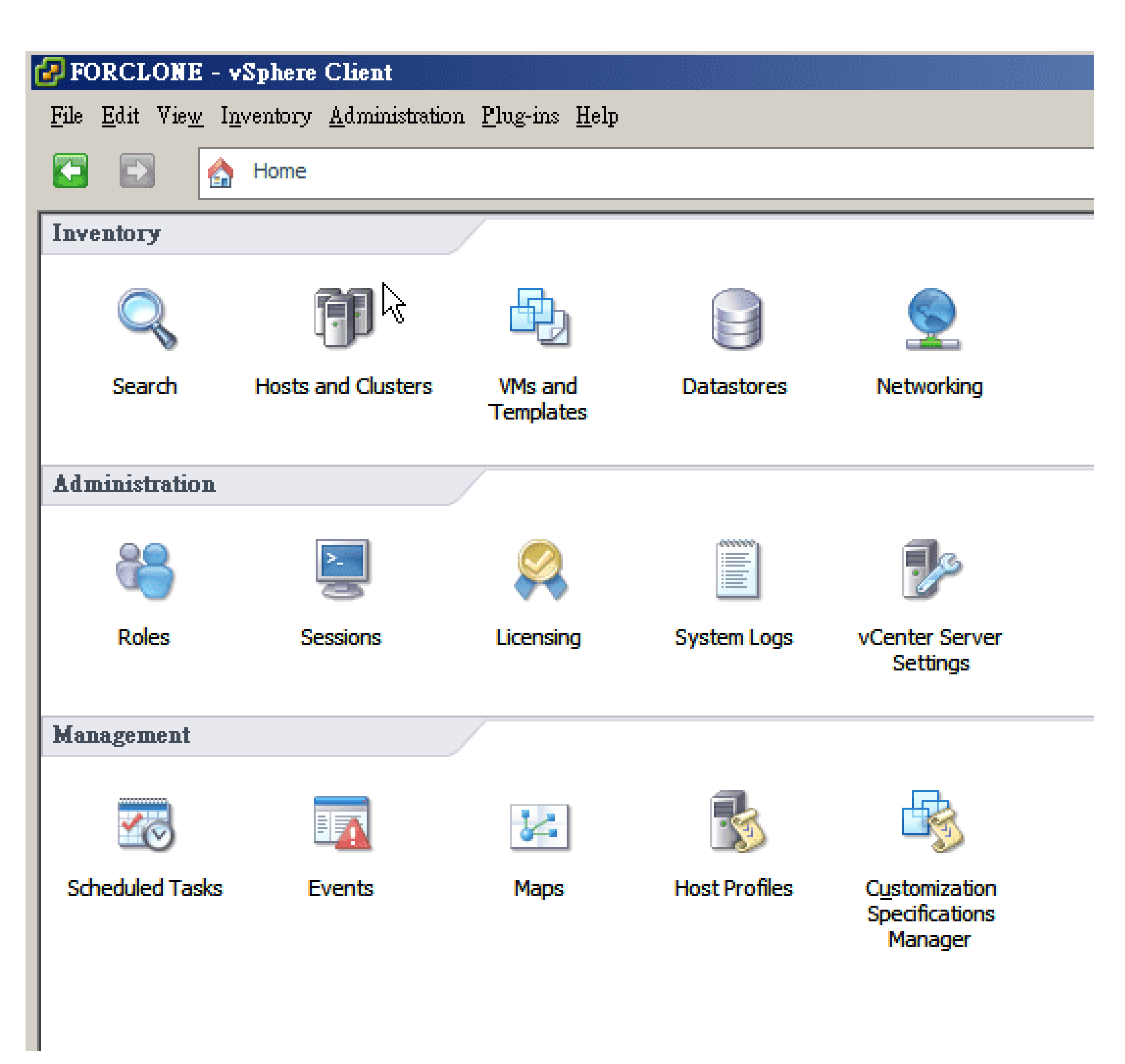
▲ 存储设备也使用多界面卡

14.5 高级的 vCenter 操作

在上一章中，我们使用了 vCenter 来管理不同的 ESX/ESXi 主机。事实上，vCenter 的功能强大，除了能管理主机之外，当然也能管理不同的 vSphere 部件。在这些部件功能排列组合之后，可以更方便地管理整个 vSphere 的主机。

14.5.1 理解更多的 vCenter 界面

通常我们在进入 vCenter 时是直接检视 Host and Virtual Machines 的视图，但在 vSphere 中，有许多其他的部件也很常用，当选择 Home 选项卡时，就会弹出部件，一共有 Inventory、Administration 及 Mangement 三大类，下面是列表：



▲ vCenter 的三大类部件，最重要的就是 Inventory

▶ vSphere 中常见的部件

Inventory 类：

- Search：可以直接查找 vSphere 中的对象。
- Host and Clusters：这是一进入 vCenter 就可看到的视图，在上一章已有说明。
- VMs and Templates：虚拟机和虚拟机样板，下一小节会有样板的实践。
- Datastores：数据仓库，就是 vSphere 中的存储设备。
- Networking：包括 ESX 主机上的网络，还有跨网段的 vNetwork，本书稍后会有说明。

Administration 类：

- Roles：角色的管理，在 vSphere 中可以给定不同的用户运行不同的任务。
- Sessions：当前正在进行的任务。
- Licensing：当前这个 vSphere 环境中主机的授权功能。
- System Logs：系统的日志，当有疑义时，检视这里的日志十分有用。
- vCenter Server Settings：配置 vCenter 的选项，这里一般不太需要动到。
- vCenter Service Status：当前 vCenter 服务的状况，也不需要动到。

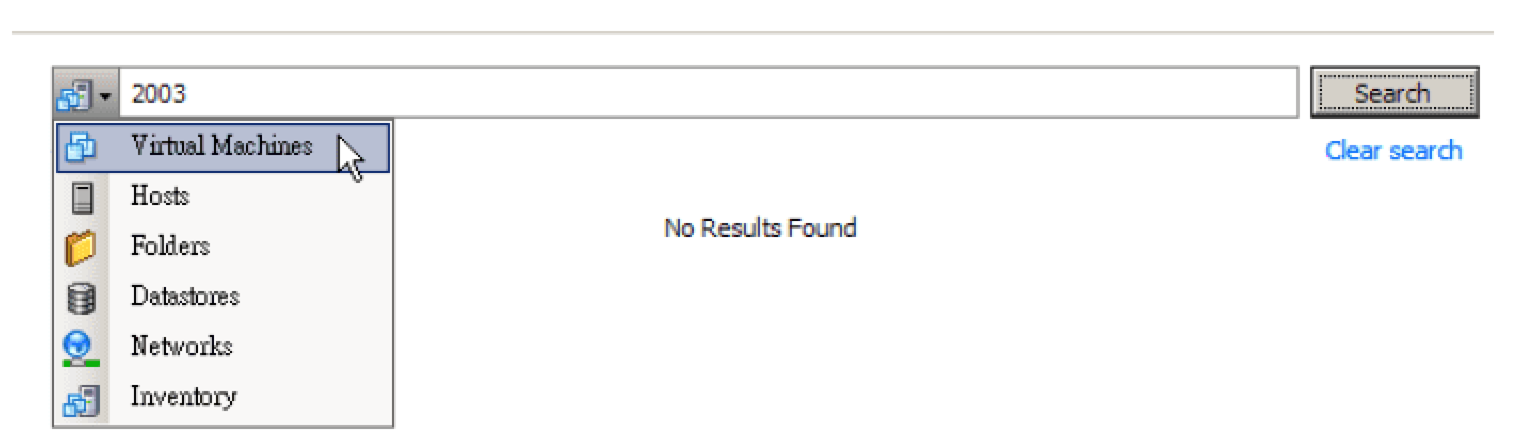
Management 类：

- Scheduled Tasks：vSphere 也可以给定排定任务，这对自动化很有帮助。
- Events：当前运行的事件。
- Maps：就是整个 vSphere 的各个部件之间的视图，上一章已经有介绍。
- Host Profiles：这是非常实用的功能，可以将某一个主机或 Cluster 的配置导到另一台上。
- Customization Specifications Manager：这是安装 VM 时可以根据不同的操作系统键入客制化的参数，如 Windows 的用户、序号等。

接下来我们就来看看最常见的操作。

1. Serach

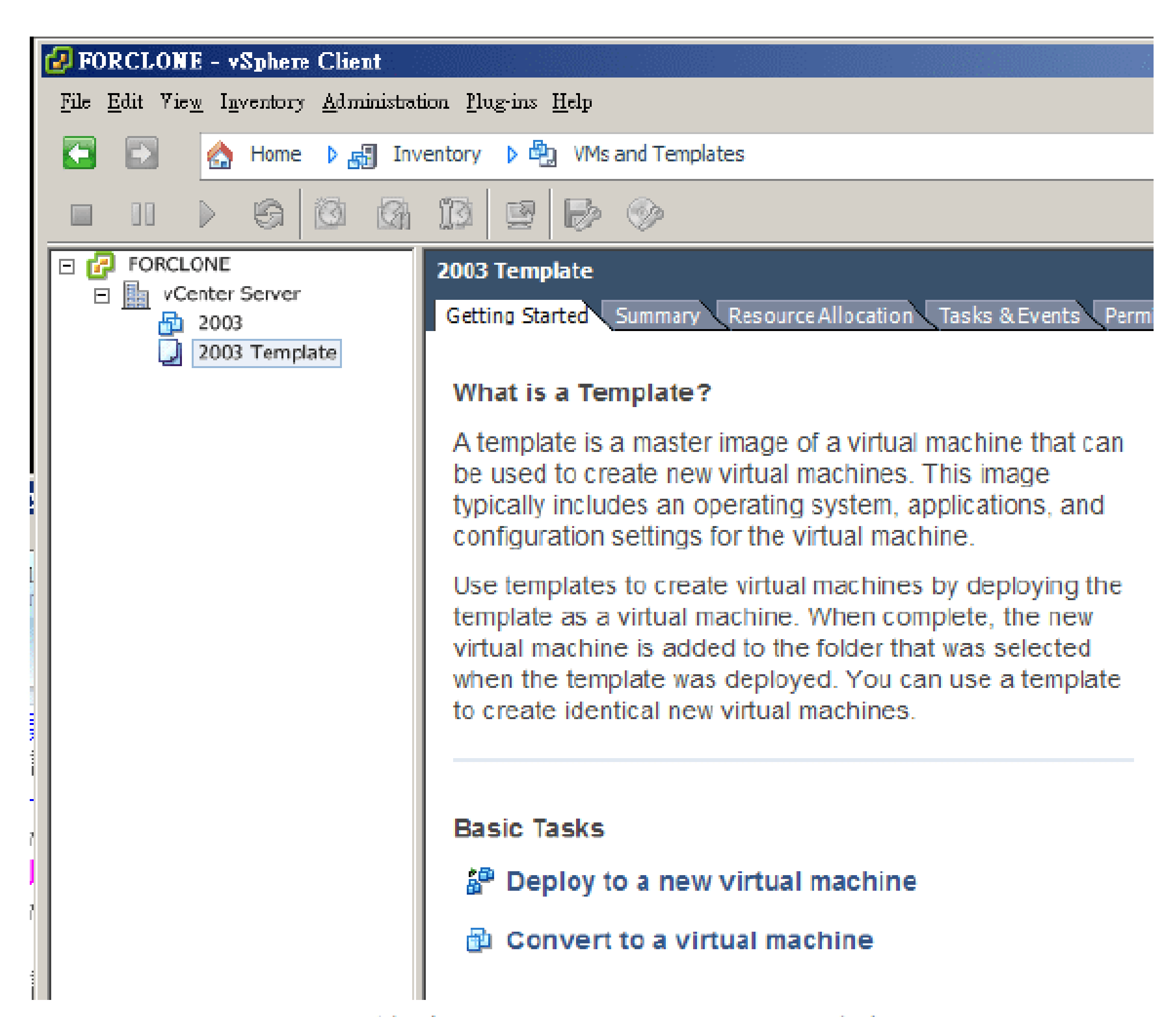
Search 是 vSphere 中最重要的功能之一。在一个 vCenter 中，如果你管理的 Datacenter、Cluster、Host 及 VM 非常多的话，你必须用这个功能来查找需要的对象。在 Serach 就可以直接查找整个 vCenter，或是根据需要的对象来查找。



▲ 可以用不同的对象来查找

2. VMS and Templates

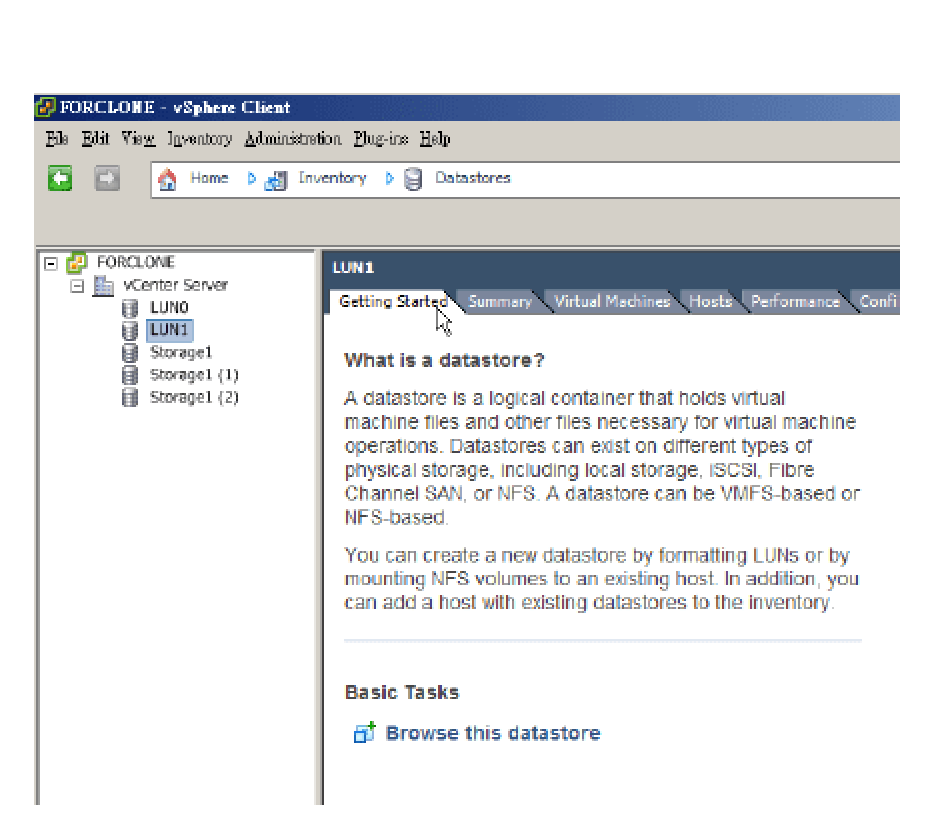
VM 不说，Templates 是一个 vCenter 环境中很重要的部件。我们可以从 Templates 直接生成虚拟机，省掉创建虚拟机的麻烦。一个 Templates 通常也是一台虚拟机创建出来的，在需要大量生成 VM 的场合中（如虚拟桌面环境），Templates 可帮助系统管理员省下大量时间。



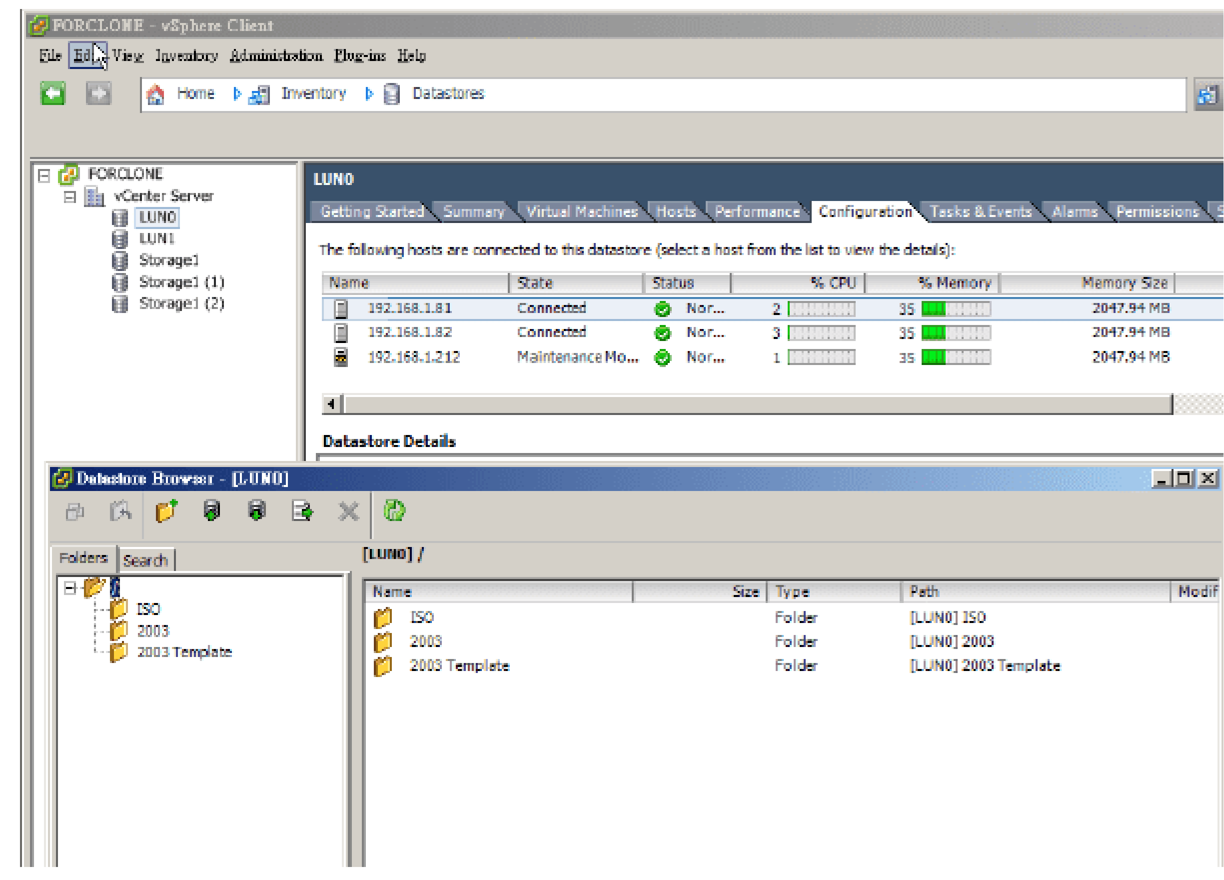
▲ 只给出 VM 及 Templates 的视图

3. Datastores

Datastores 可以将整个 vCenter 的 Datastores 统一管理，不管是 NFS 还是其他的 iSCSI 或本地存储，都可以在这里管理。



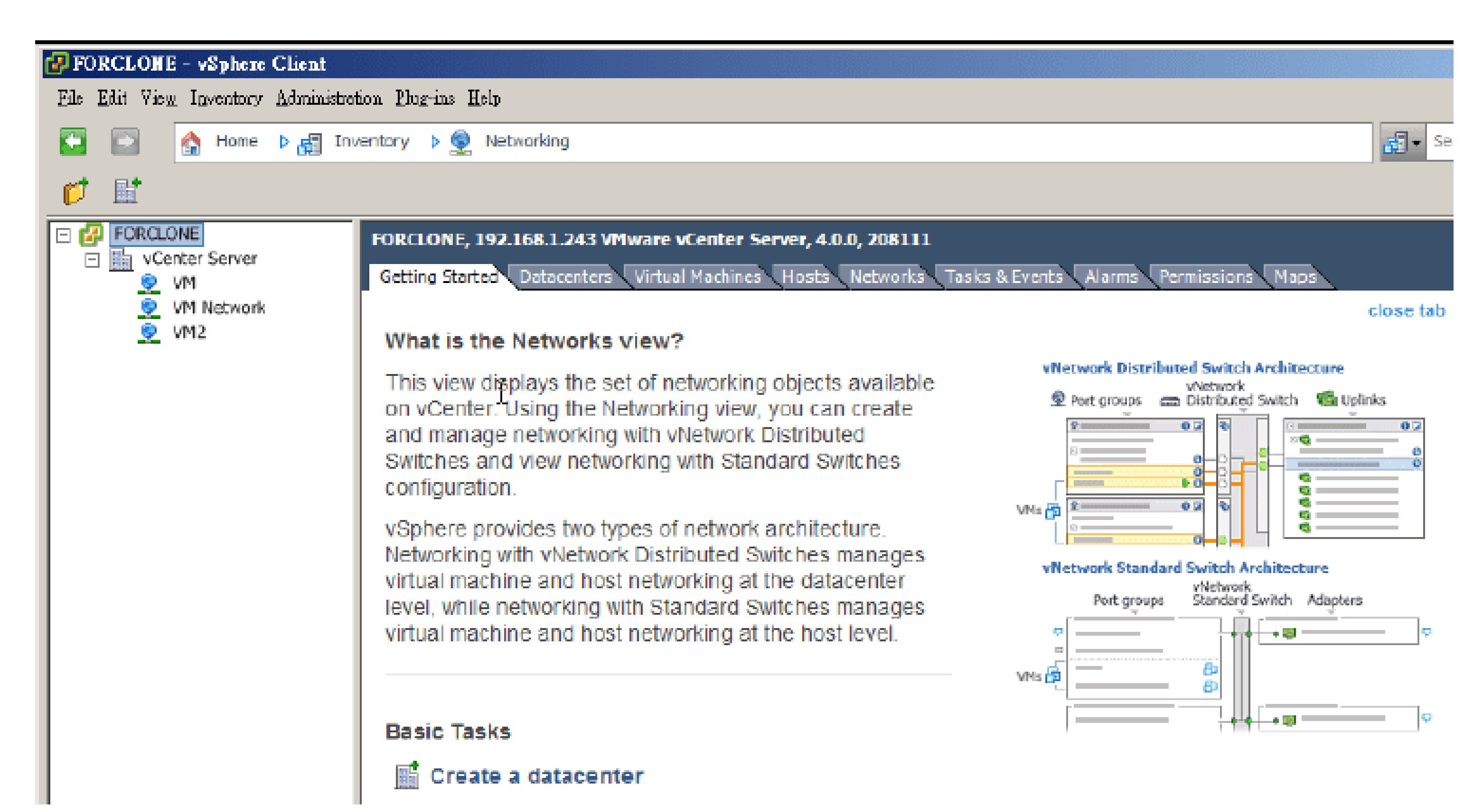
▲ Datastores 是所有存储的基础



▲ 可以直接浏览其中的内容

4. Networking

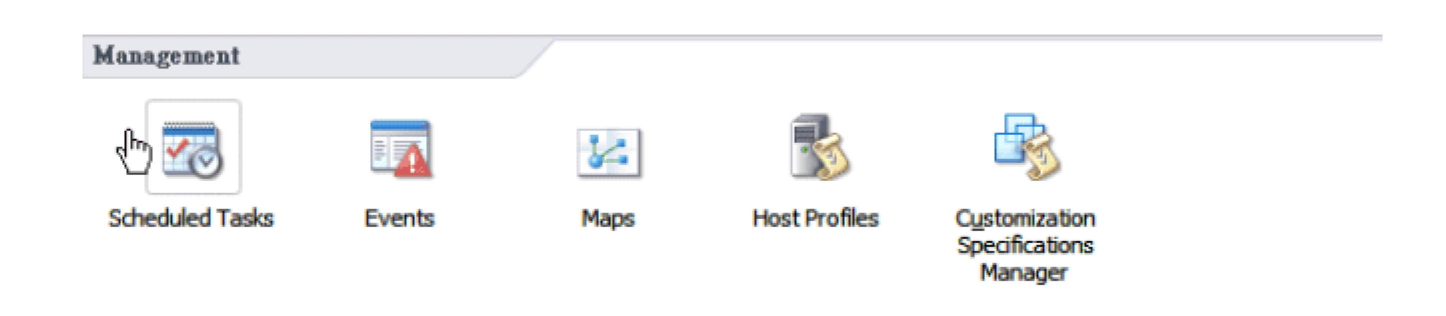
这里管理了整个 vCenter 的网络，包括单机的 vSwitch 和跨主机的 vNetwork，本书稍后会有完整的 vNetwork 及 Distributed vSwitch 的实践。



▲ 所有主机上的网络都会在此给出

5. Host Profiles

当我们辛辛苦苦创建了一个 Datacenter/Cluster/Host 的各式各样环境之后，如果还要再创建一个一模一样的克隆，是不是需要重新再来一次？只要使用 Host Profiles 的功能之后，你可以把一台 ESX 主机的整个网络或其他的配置做成一个主机样板，本章稍后会有说明。



▲ Host Profiles 十分好用

14.5.2 理解 Template VM

vSphere 中有一个 Template 功能，允许你将 VM 克隆到一个模板中。当模板创建完毕之后，

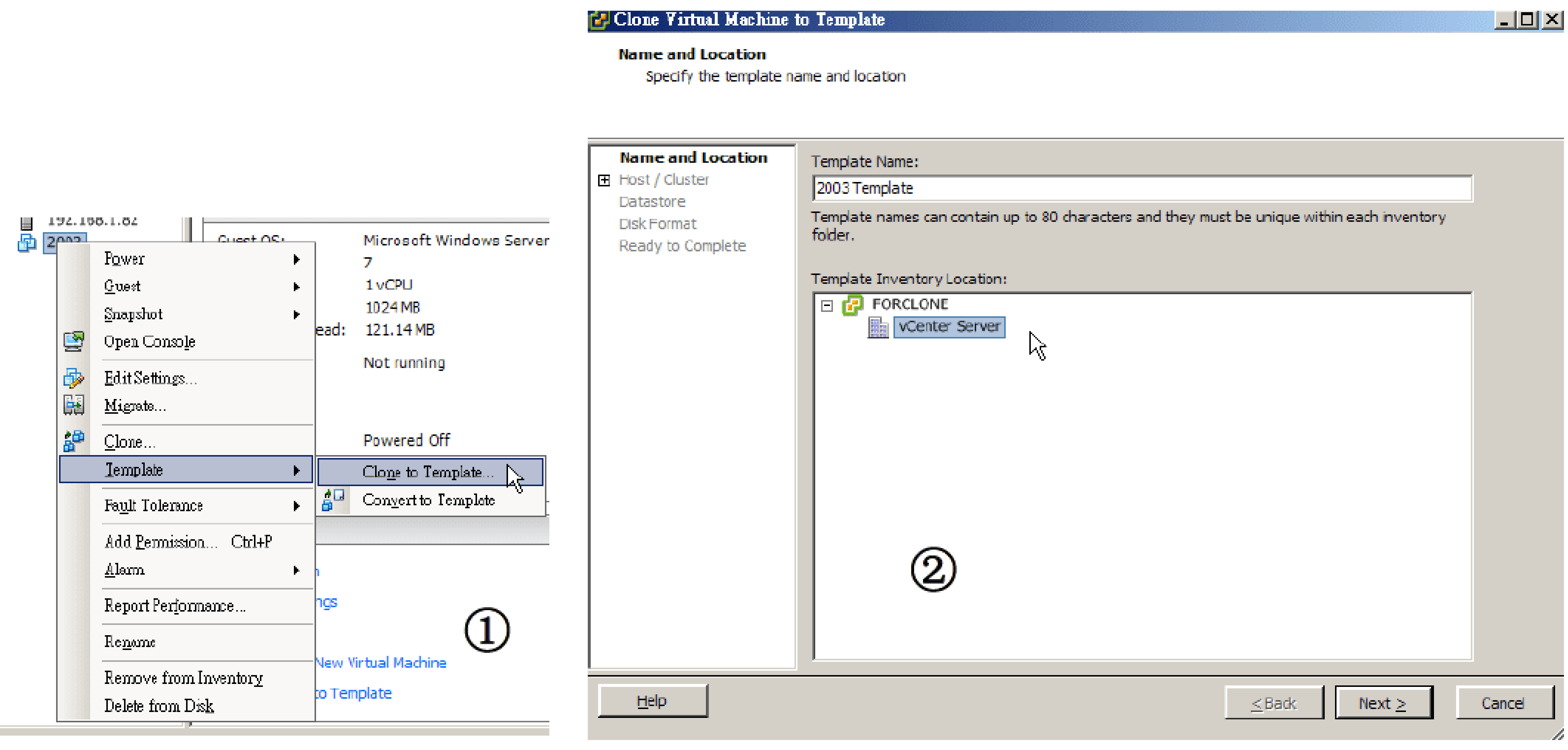
你可以从这个模板快捷生成 VM，我们就来看看模板的使用。

1. 从 VM 创建 Template

你可以将 VM 克隆或转换成 Template，两者的差别就是克隆后，这个 VM 还可以继续使用，但如果是转换后，这个 VM 以后只能用来生成新的 VM 了。

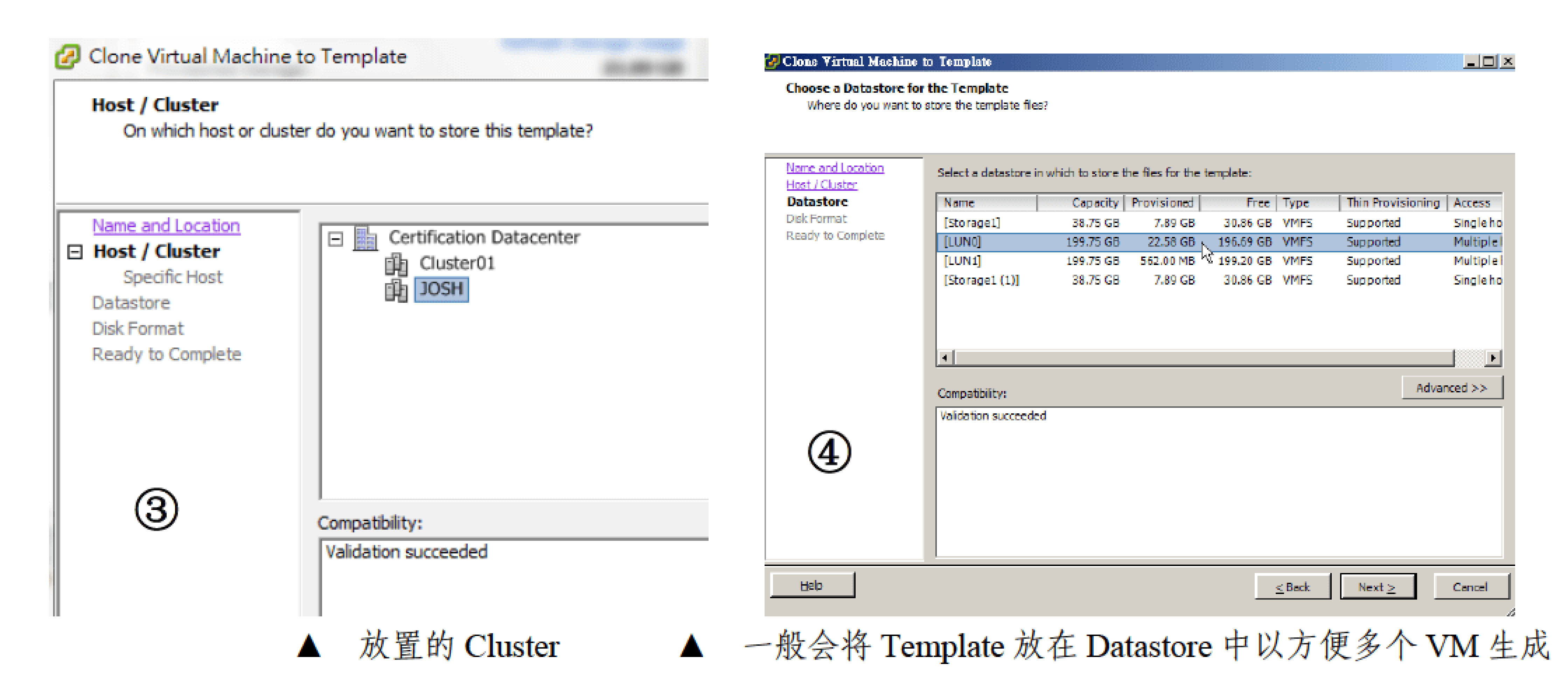
▶ 将 VM 转换成 Template

1. 选择你要转换的 VM，在其上右击，选择 Template/Clone to Template 选项。记住此时 VM 是要在关机状态。
2. 接下来是 Template 名称以及放置的地方，我们就取名 2003 Template，并且放在原来的 Datacenter 中。单击 Next 按钮继续。



▲ 选择选项　　　　▲ 选择名称及放置地点

3. 接下来是存储的 Cluster，单击 Next 按钮继续。
4. 接下来是存放的 Datastore。我们就放在网络存储 LUN0 中方便其他主机的使用。单击 Next 按钮继续。

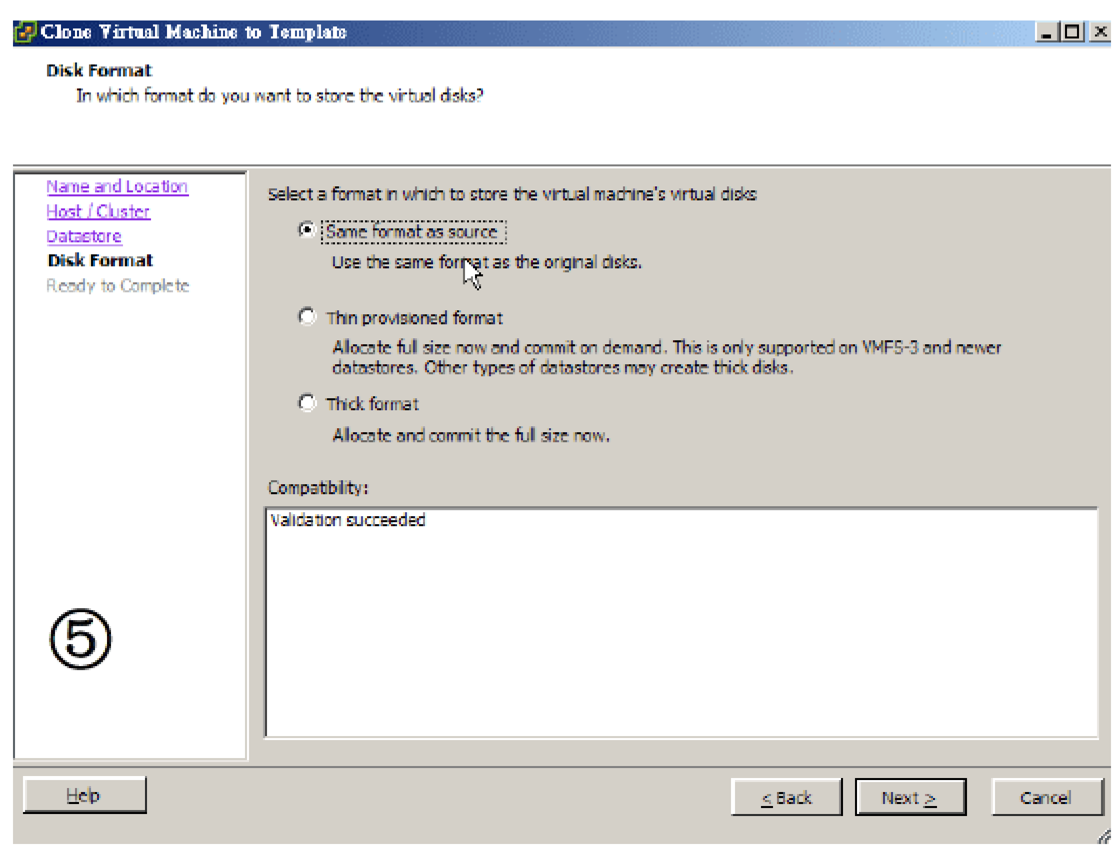


Name	Capacity	Provisioned	Free	Type	Thin Provisioning	Access
[Storage1]	38.75 GB	7.89 GB	30.86 GB	VMFS	Supported	Single ho
[LUN0]	199.75 GB	22.58 GB	196.69 GB	VMFS	Supported	Multiple
[LUN1]	199.75 GB	562.00 MB	199.20 GB	VMFS	Supported	Multiple
[Storage1 (1)]	38.75 GB	7.89 GB	30.86 GB	VMFS	Supported	Single ho

▲ 放置的 Cluster　　　　▲ 一般会将 Template 放在 Datastore 中以方便多个 VM 生成

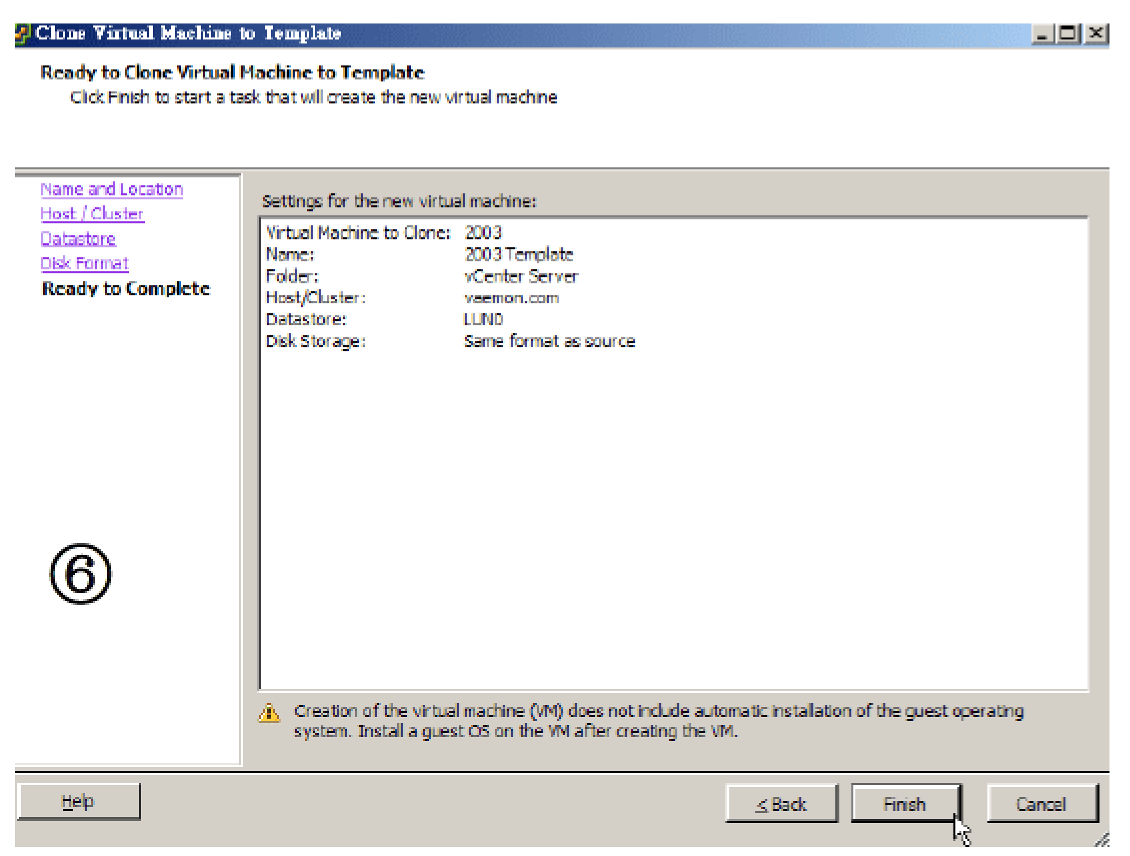
5. 接下来是存放这个 Template 的硬盘方式。一般来说，我们会选择和来源 VM 一样的格

式较为方便。单击 Next 按钮继续。



▲ 硬盘的结构，保持和原来一样

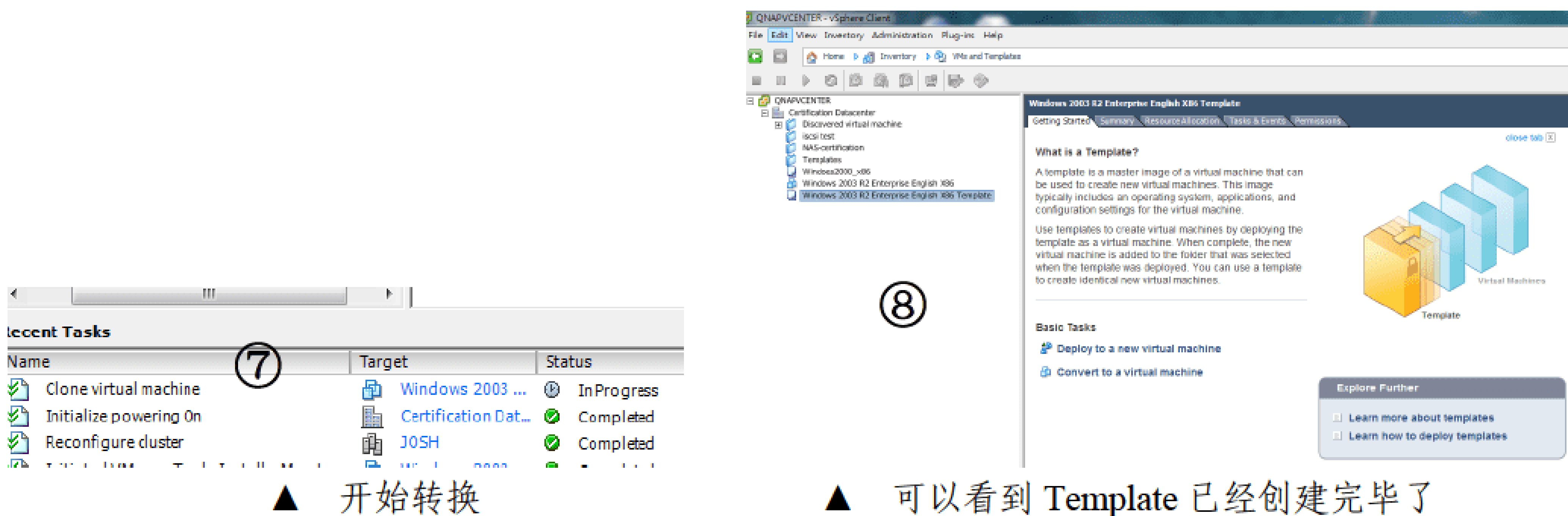
6. 接下来是总结画面，单击 Finish 按钮继续。



▲ 总结画面

7. 接下来会开始转换，视这个 VM 的大小，会花上数分钟到数十分钟不等。

8. 当弹出如下图所示画面时，意味着这个 Template 已经创建完毕了。



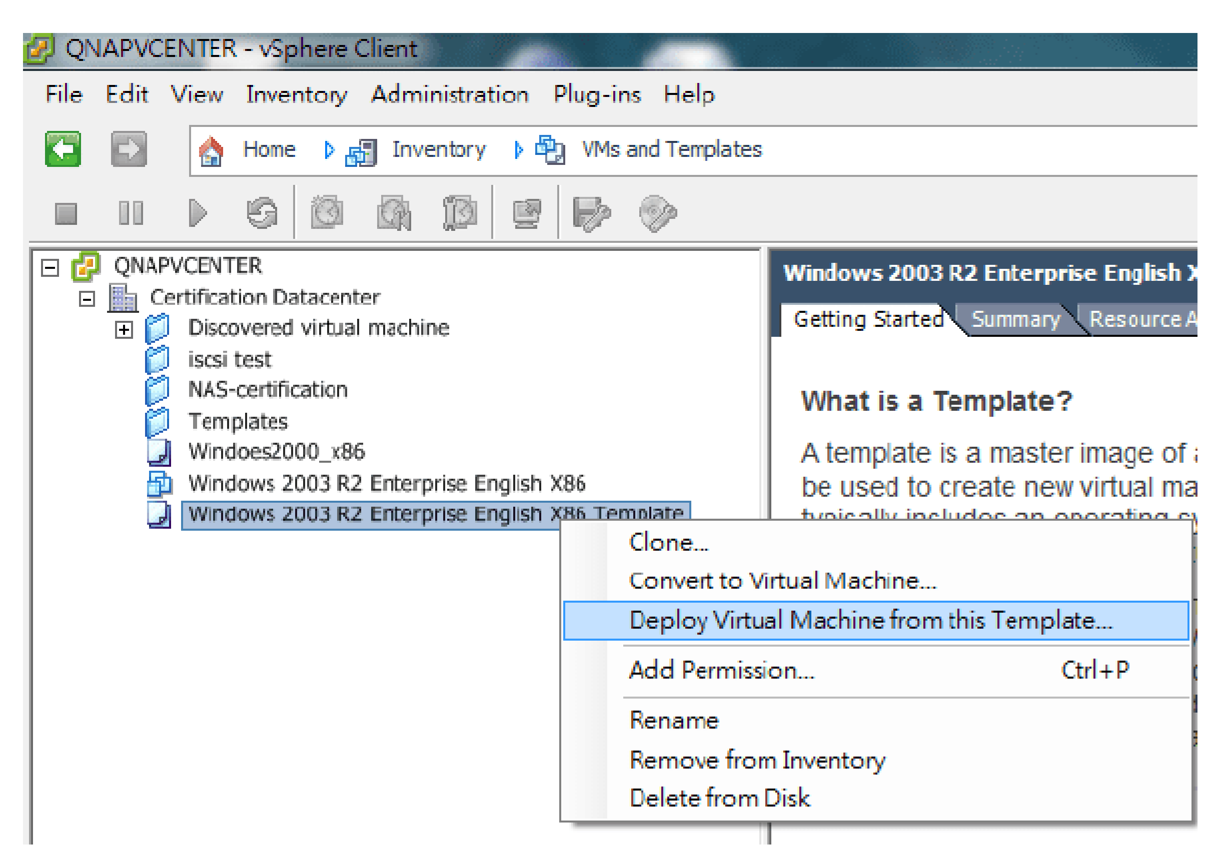
▲ 开始转换 ▲ 可以看到 Template 已经创建完毕了

2. 从 Template 创建 VM

从 Template 创建 VM 也十分容易，我们就来看看。

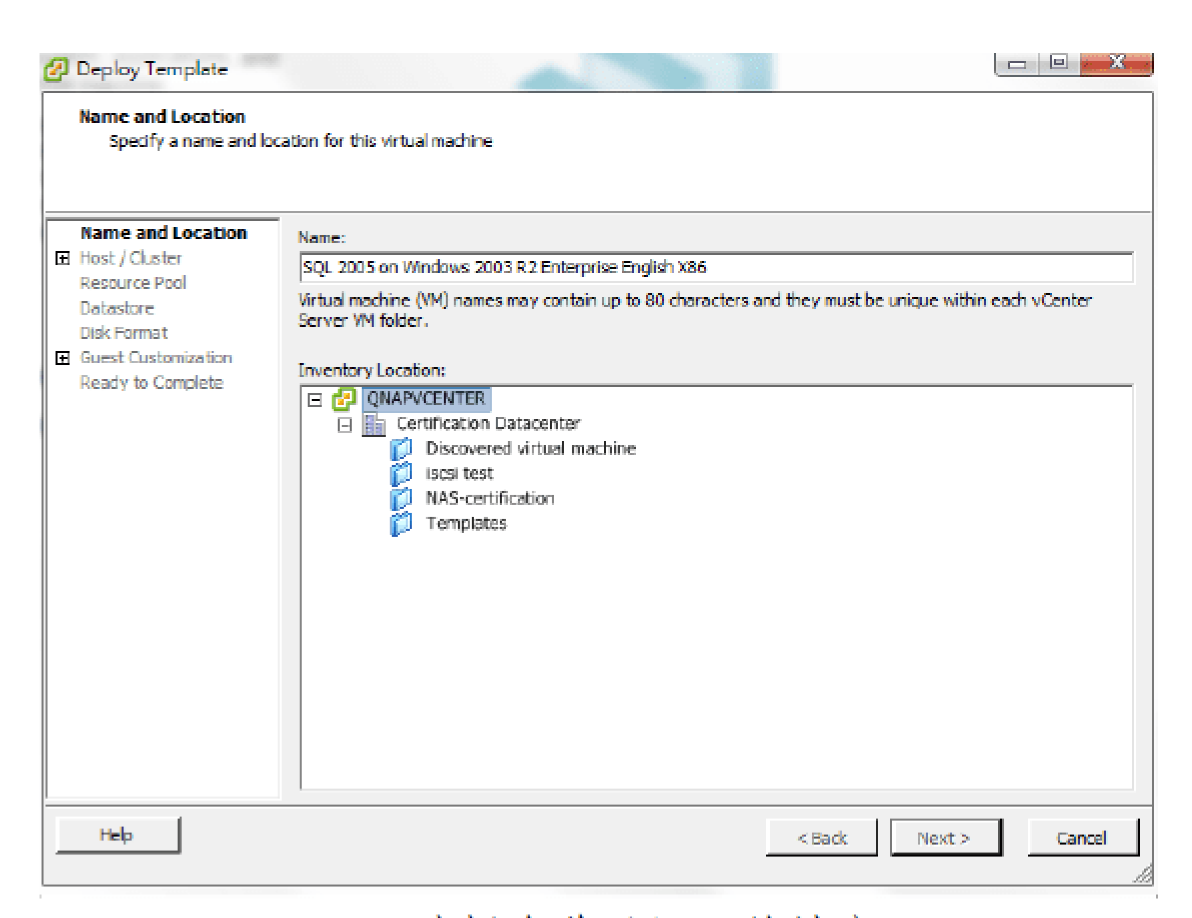
▶ 从 Template 创建 VM

1. 在要创建的 Template 上右击，选择 Deploy Virtual Machine from this Template 选项。



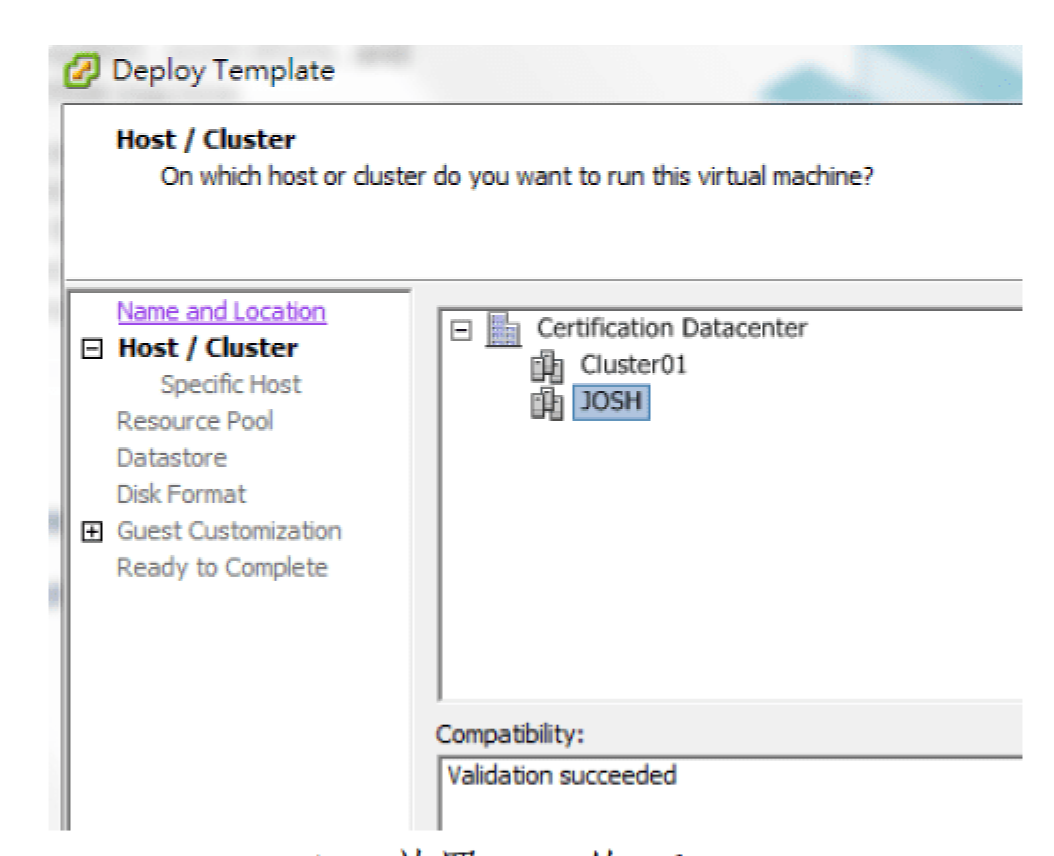
▲ 选择创建 VM

2. 接下来键入新的 VM 的名称，并且选择要放置新的 VM 的地方，单击 Next 按钮继续。



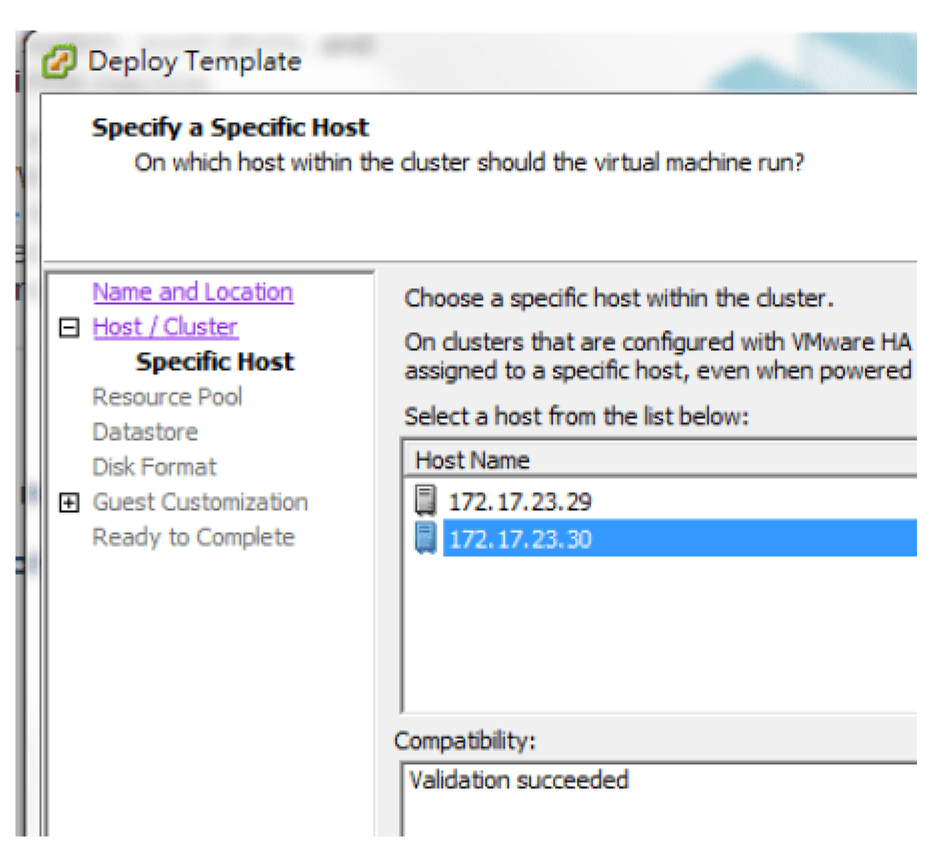
▲ 选择名称及压入的地方

3. 接下来选择要放置的 Cluster，单击 Next 按钮继续。



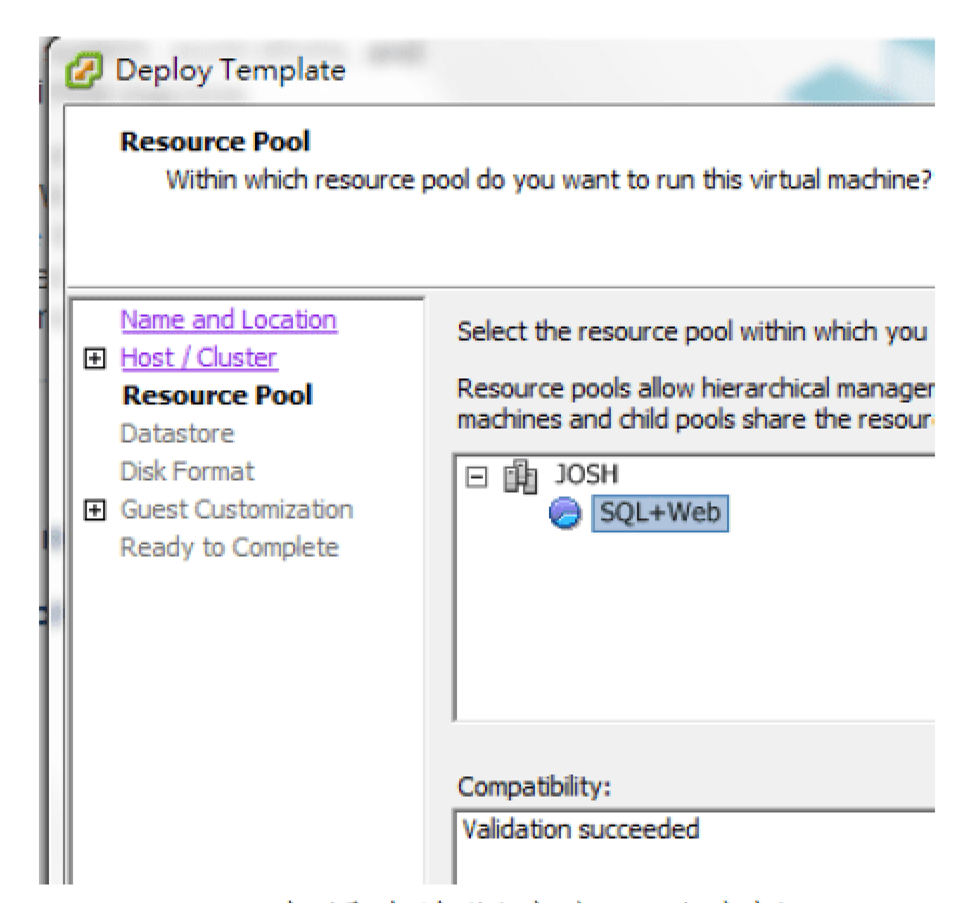
▲ 放置 VM 的 Cluster

4. 接下来选择要放的主机名，单击 Next 按钮继续。



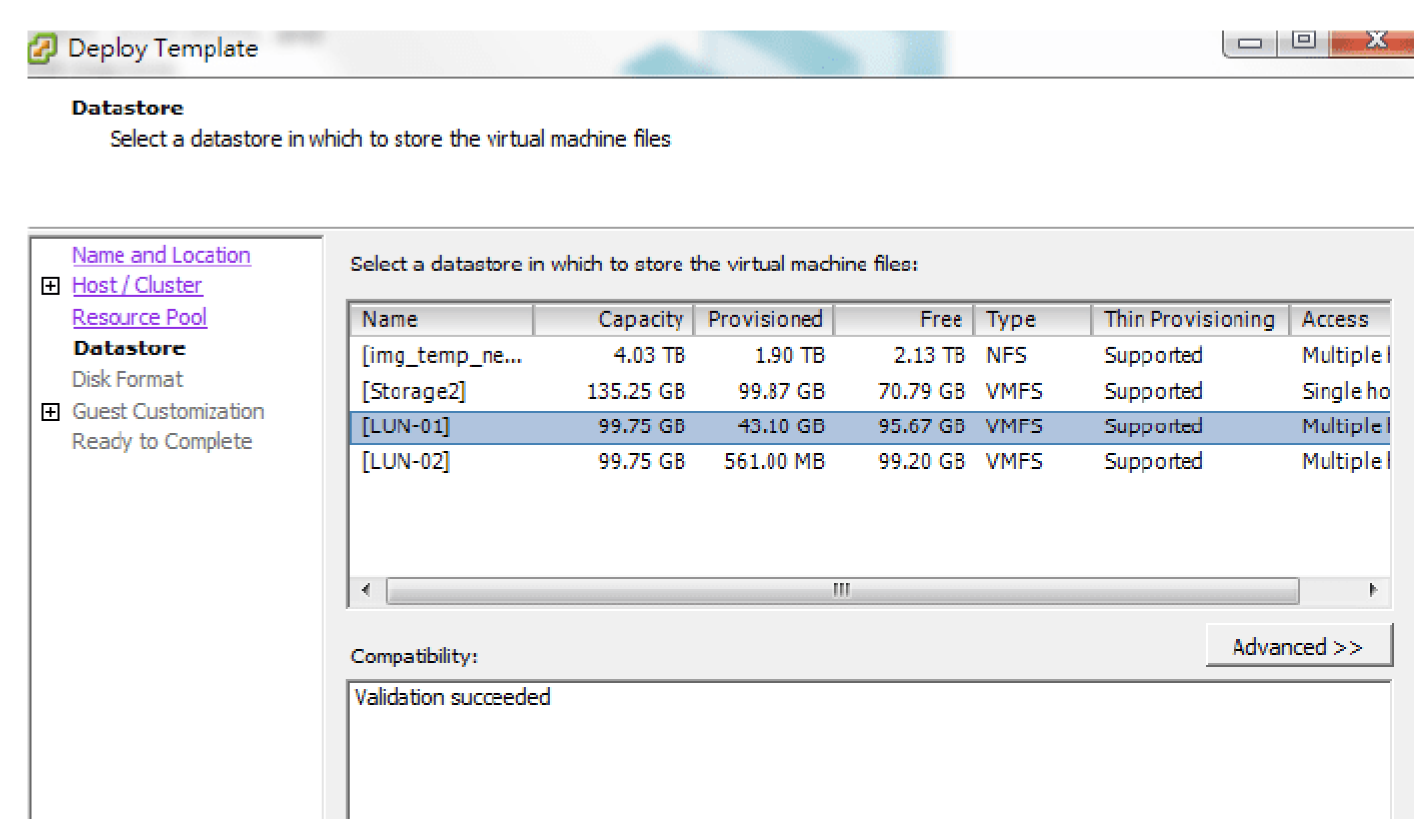
▲ 放置 VM 的主机

5. 如果你的 Cluster 有使用 Resource Pool，也要选择。单击 Next 按钮继续。



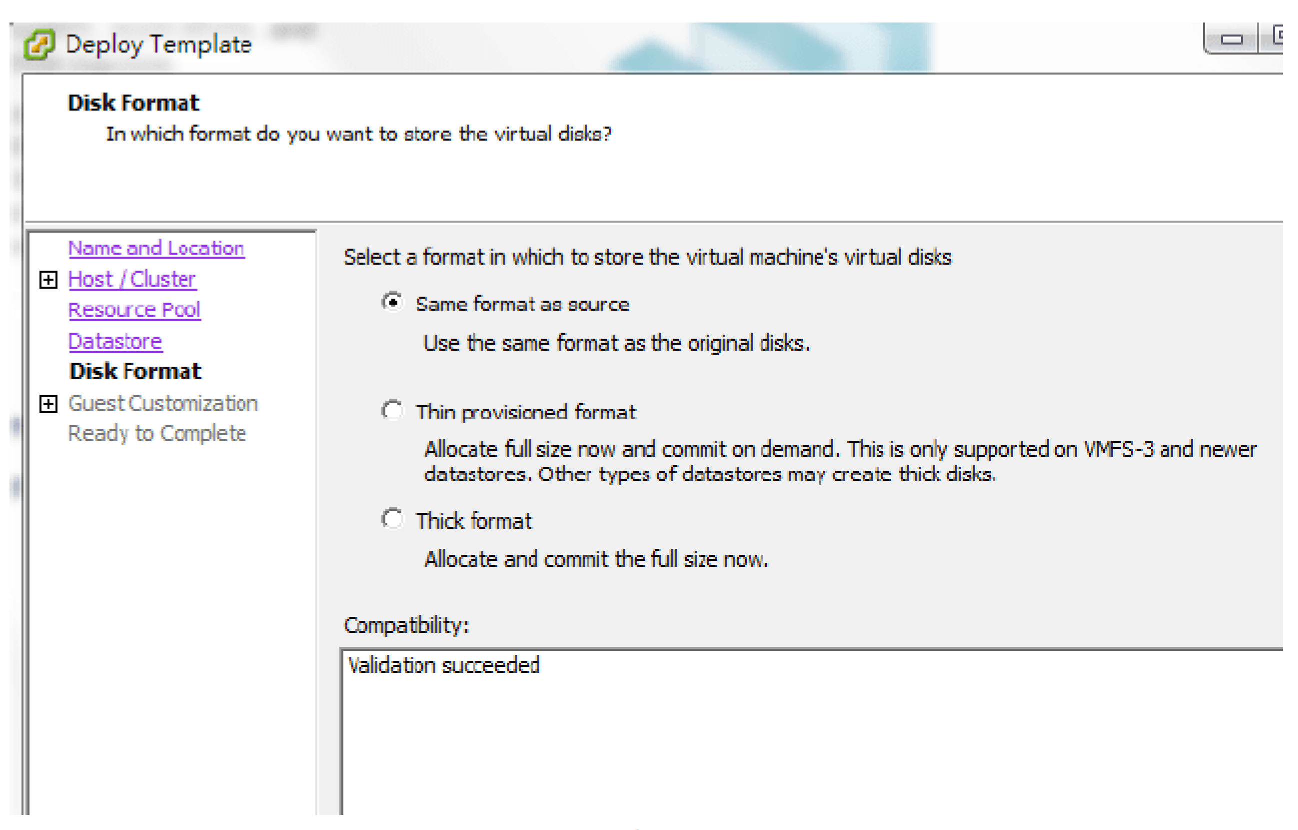
▲ 如果有资源池也可以选择

6. 接下来是选择 VM 的置放 Datastore，单击 Next 按钮继续。



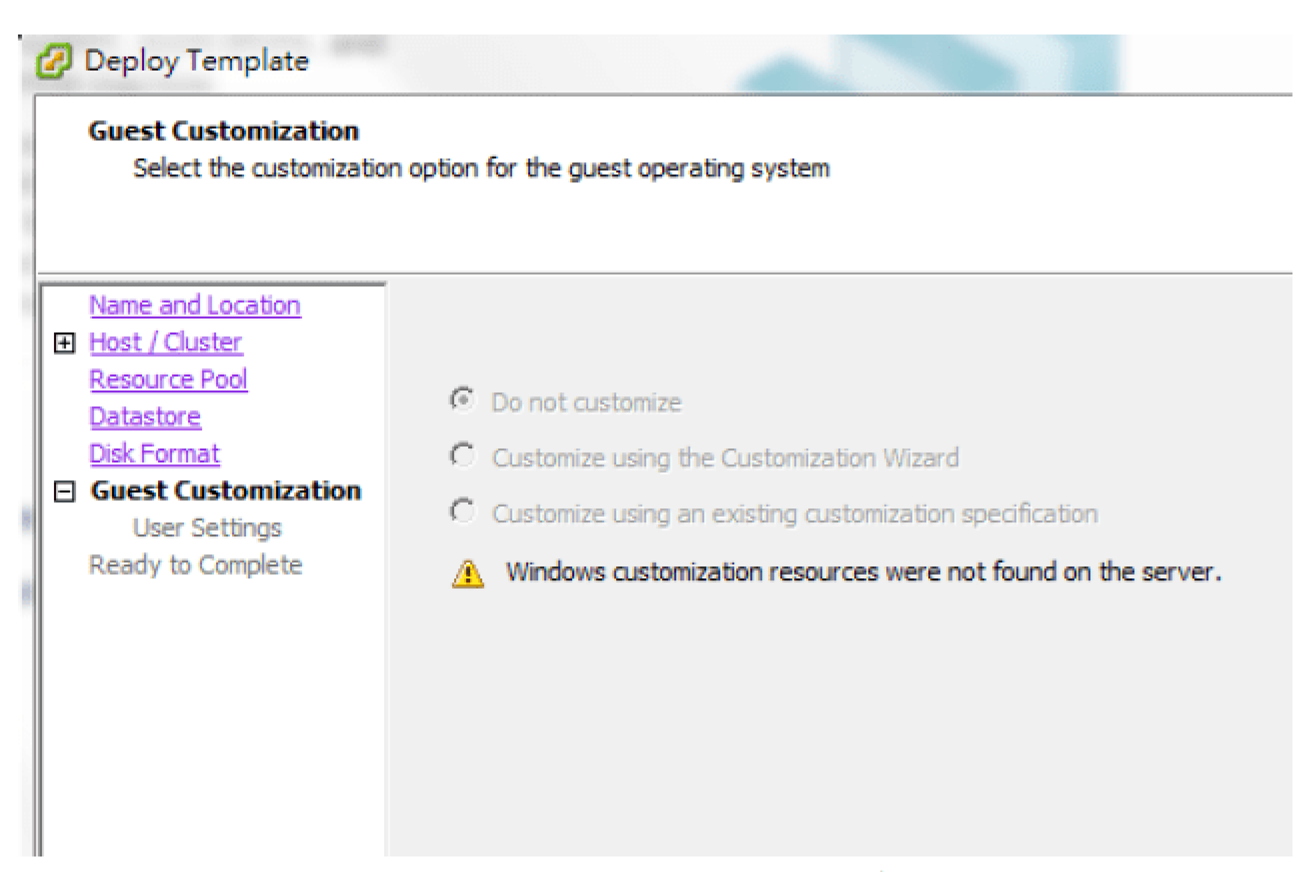
▲ 放置 VM 的数据仓库

7. 接下来是选定新的 VM 的磁盘格式，也是以默认值为主。单击 Next 按钮继续。



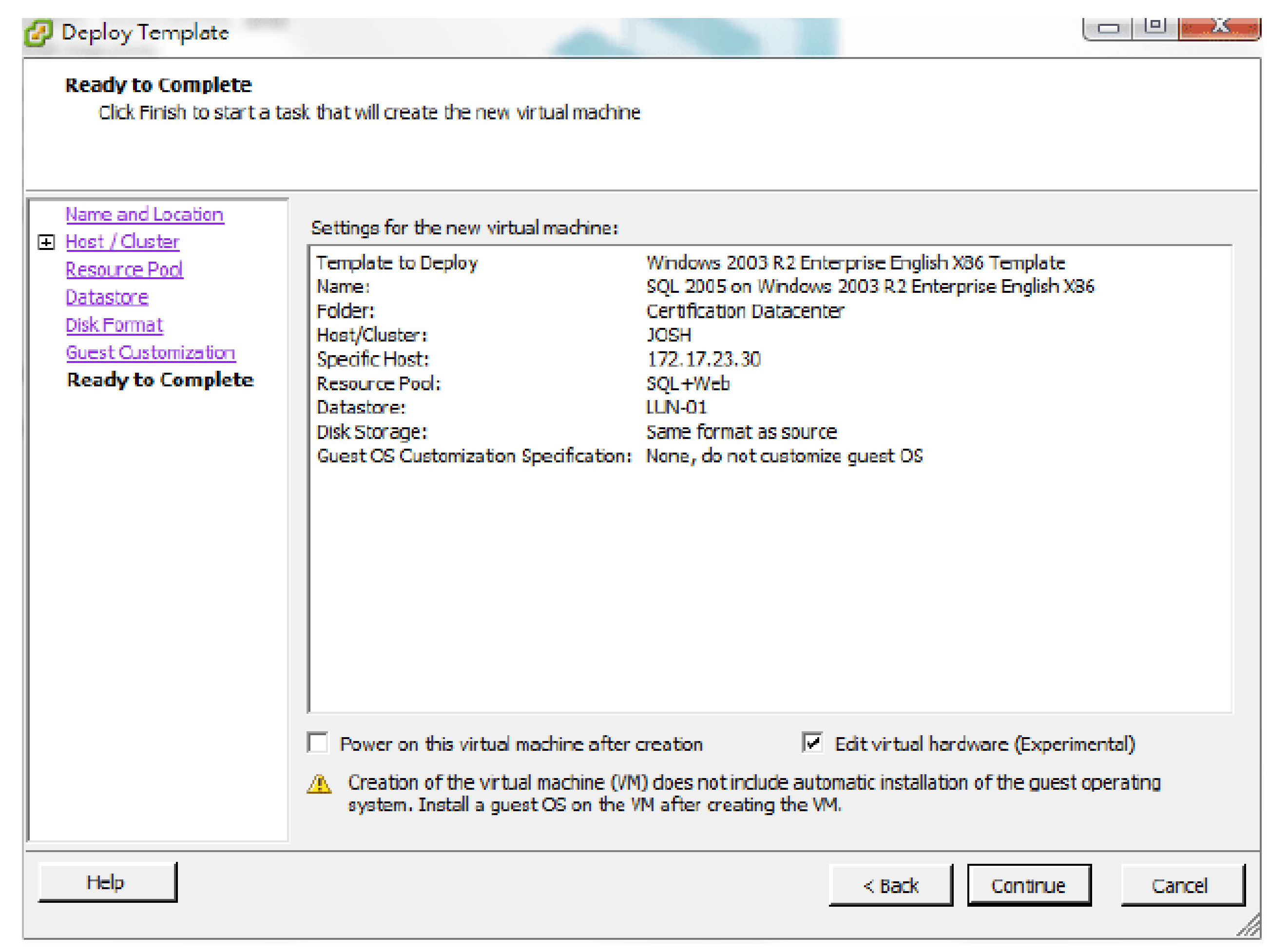
▲ 选择创建的格式

8. 在创建新的 VM 时，系统会询问要不要创建定制化安装条件，笔者建议不要，因为会让整个流程更麻烦。单击 Next 按钮继续。



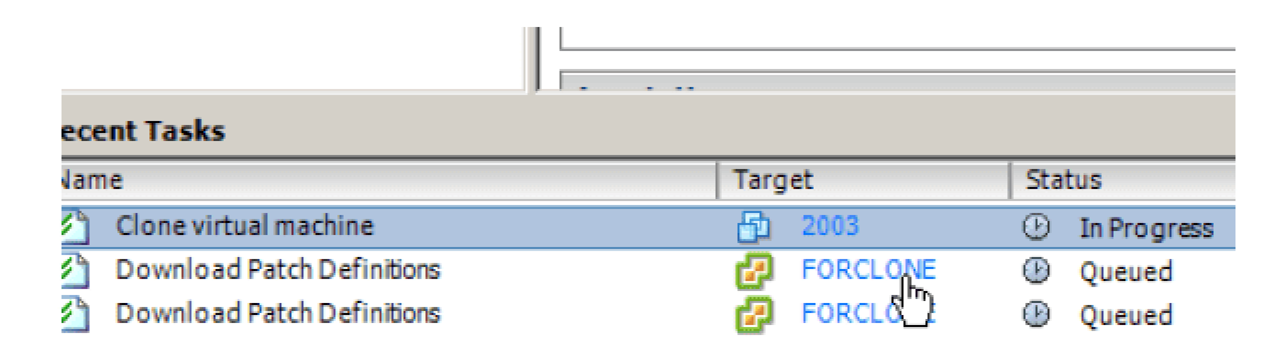
▲ 定制化在大量制造时较有用，一般创建时会比较麻烦

9. 接下来是总结画面，确定之后可以单击 Continue 按钮继续。你也可以选择编辑新的 VM 的硬件配置。



▲ 总结画面

10. 接下来就会开始创建 VM 了。



▲ 开始创建 VM

14.5.3 理解 Host Profile

vSphere 允许用户将一个主机的配置存在一个 Profile 中。如果有其他主机加入时，就可以直接将这个 Profile 套用在新加入的主机上，省下大量的配置时间。一般来说，Profile 中存储的多为网络配置和存储设备的配置，其他如内存的预留、防火墙、安全等配置较为少用。在上一小节中，我们创建了 host1 的 ESX 网络配置，如果你想将这个配置套用到 host2 另一台 ESX 上，最方便的做法就是使用 Profile 了。

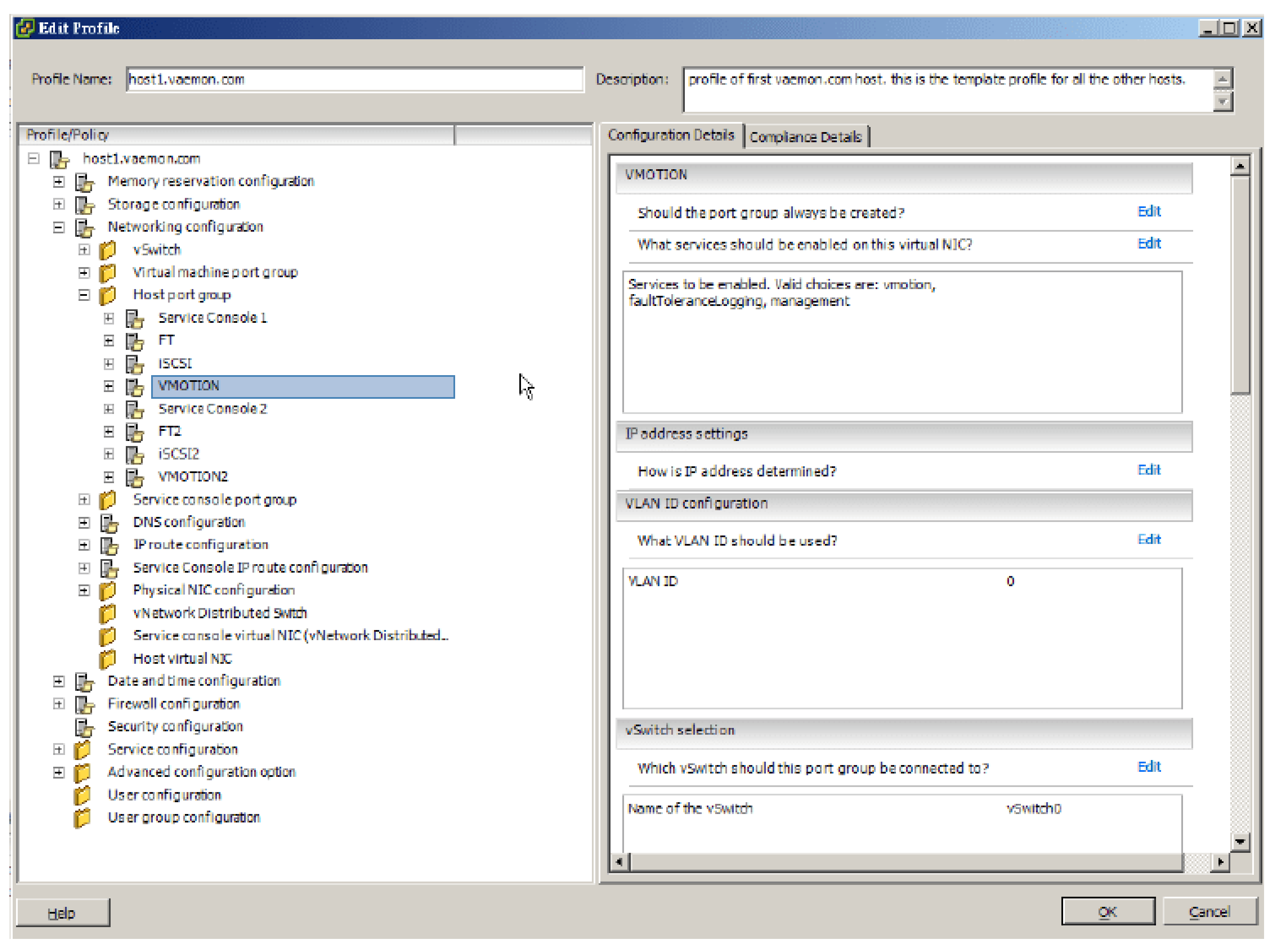
1. 操作 vSphere 的 Host Profile

当你选择 Home 选项卡，并且选择 Host Profiles，就可以进入 Host Profiles 的画面。通常我们会先将一台 ESX 主机做成完整的配置，再将这个 ESX 主机的 Profile 导出，成为模板 Profile，以后所有的新 ESX 主机就可以套用这个 Profile 了。



▲ 这是 Host Profile 的画面

由于每一个 ESX 主机的配置还是会有细微的不同，如 IP 位置、主机名等，因此 Profile 的存储通常是一个“轮廓”，细部的具体数字还是需要用户重新键入，因此在套用 Profile 的时候，也会要求你键入这些数值。



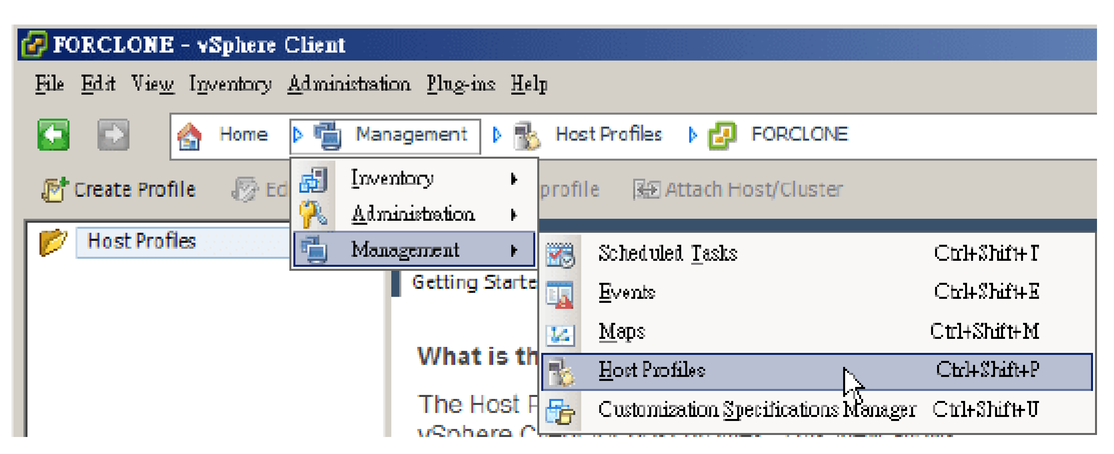
▲ 通常需要重新键入一些细部选项

2. 创建一个新的 Profile

要创建一个新的 Profile，首先我们要先创建一台 ESX 主机，并且将网络及存储设备都配置好，接下来就是导出 Profile 了。

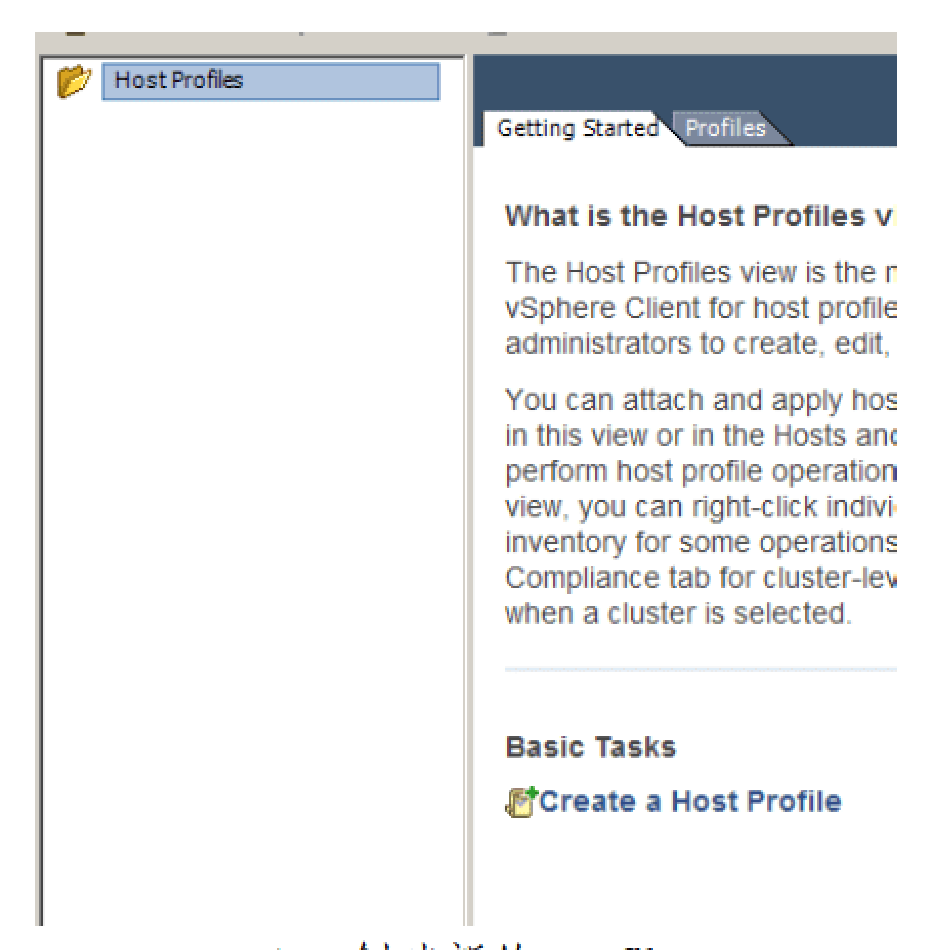
▶ 导出 ESX 主机的 Profile

1. 选择 Home 中的 Host Profiles 选项。



▲ 选择选项

2. 进入 Profile 的管理画面。选择 Create a Host Profile 链接。

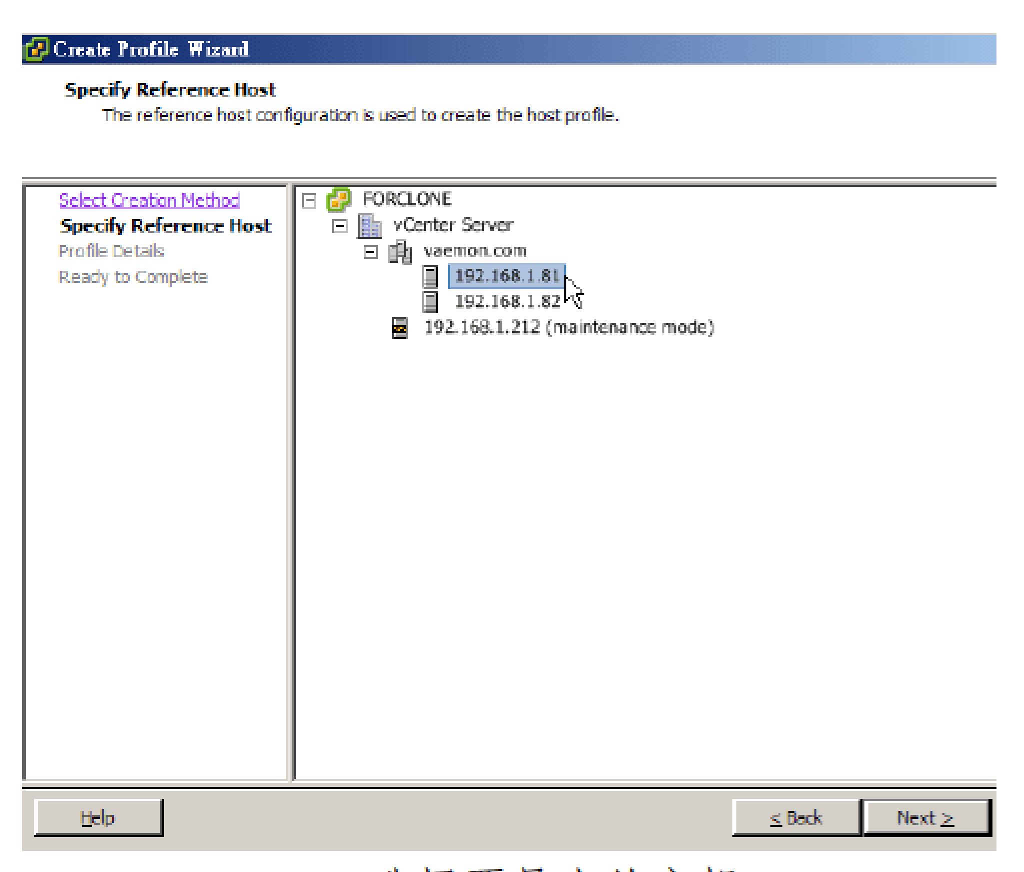


▲ 创建新的 Profile

3. 选择第一项从现有主机导入 Profile，单击 Next 按钮继续。

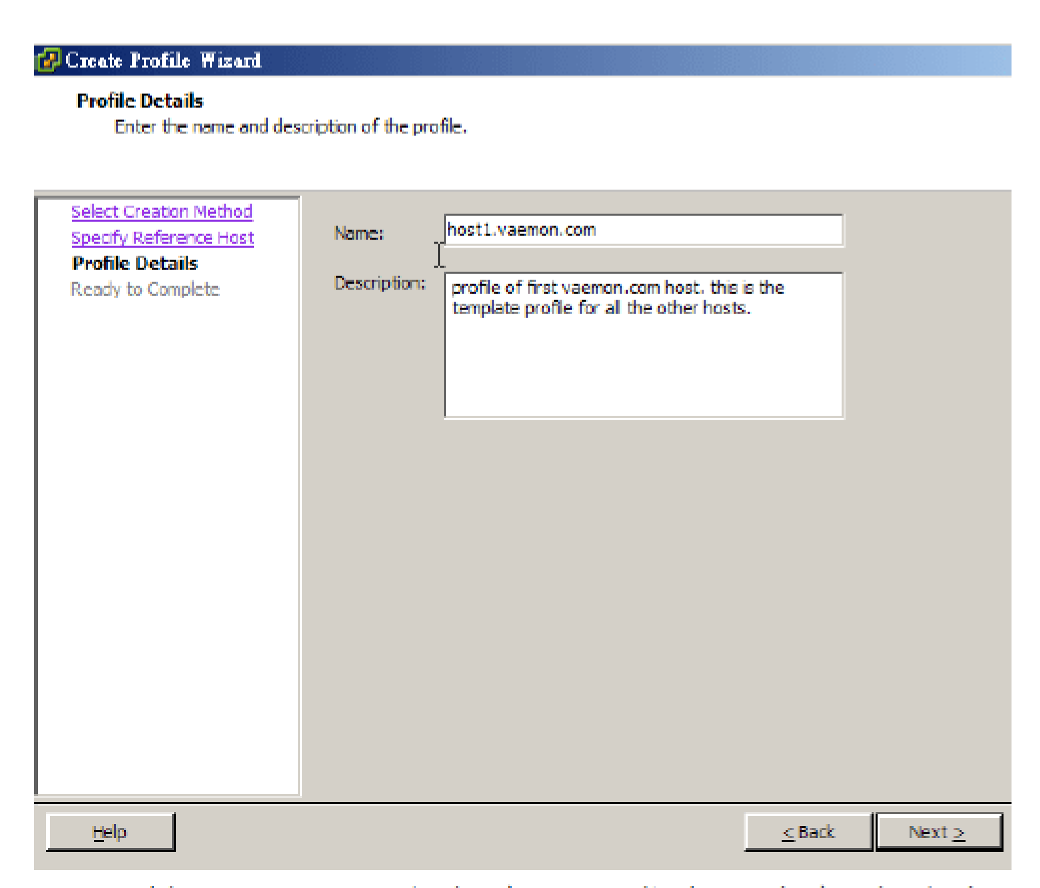
▲ 导入 Profile

4. 选择要导出的主机，单击 Next 按钮继续。



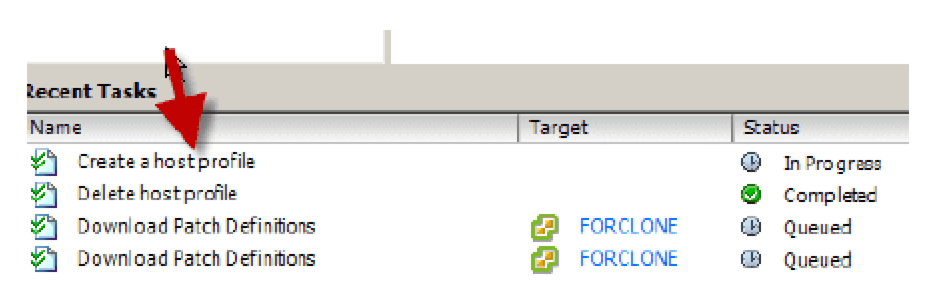
▲ 选择要导出的主机

5. 接着键入 Profile 的名称，并且键入描述，单击 Next 按钮继续。



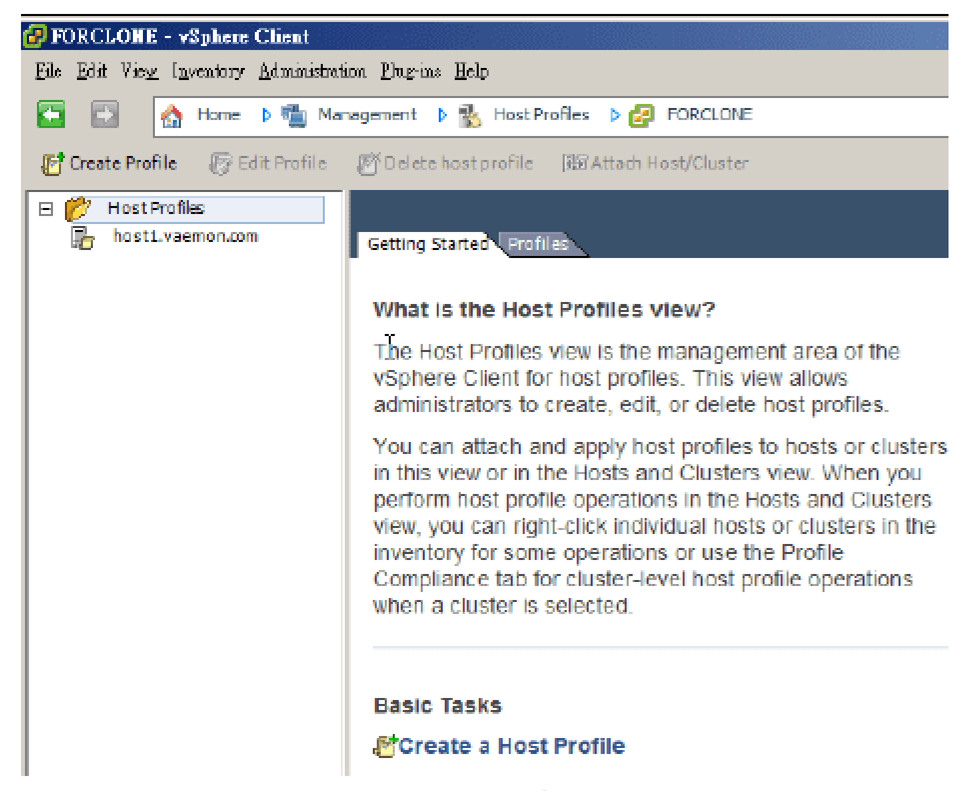
▲ 键入 Profile 的名称，通常会以主机名为主

6. 接下来会开始创建 Profile。



▲ 开始创建

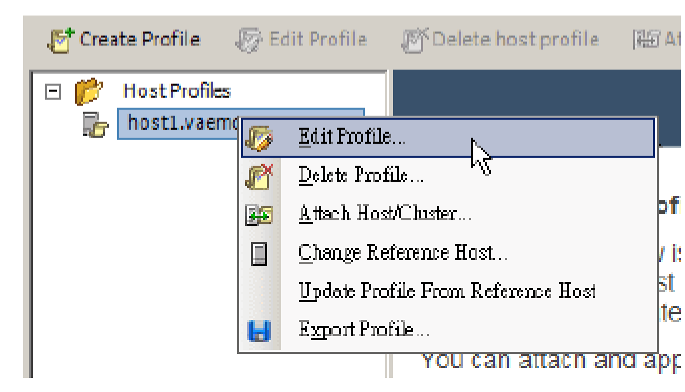
7. 当落实之后，可以看到 Profile 管理画面中已经有一个主机的 Profile 了。



▲ Profile 已经创建完毕了

3. 管理 Profile

在 Profile 管理画面右击，选择 Edit Profile 选项时，就可以管理刚才创建的 Profile。事实上管理 Profile 主要的目的还是为了给其他的主机用。但是 Profile 的细部选项还是得靠我们自身维护。

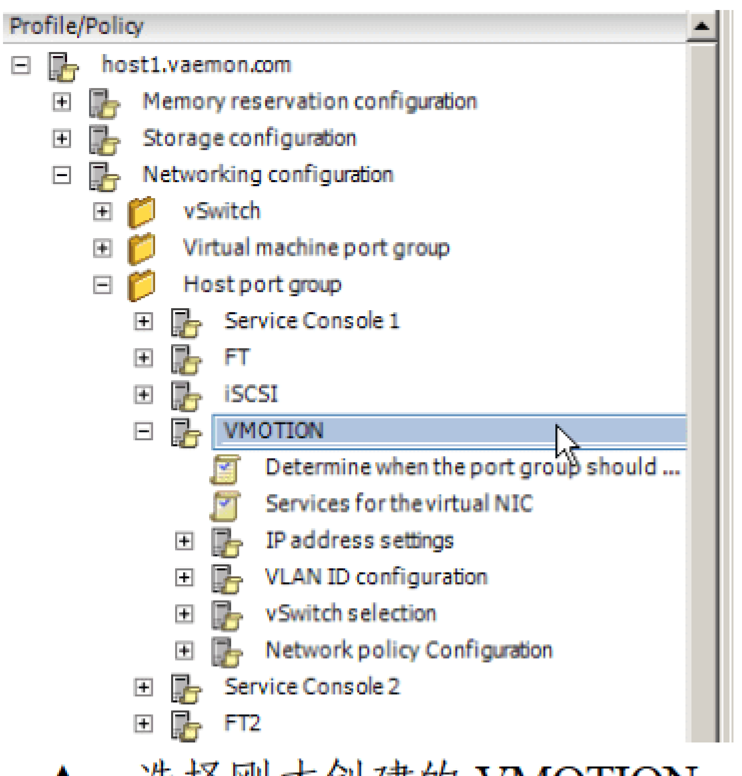


▲ 编辑 Profile

举例来说，在创建 VMkernel 的时候，我们在 host1 中创建了一个 VMOTION，为了使用 VMotion 功能。但是 Profile 的默认值是两个选项都不选中，因此必须在 Profile 创建之后，在套用在其他主机之前，我们必须先将 VMOTION 这个 VMkernel 的 VMotion 选项勾起来，让其他主机的细部配置也和 host1 一模一样。我们就举一个简单的例子。

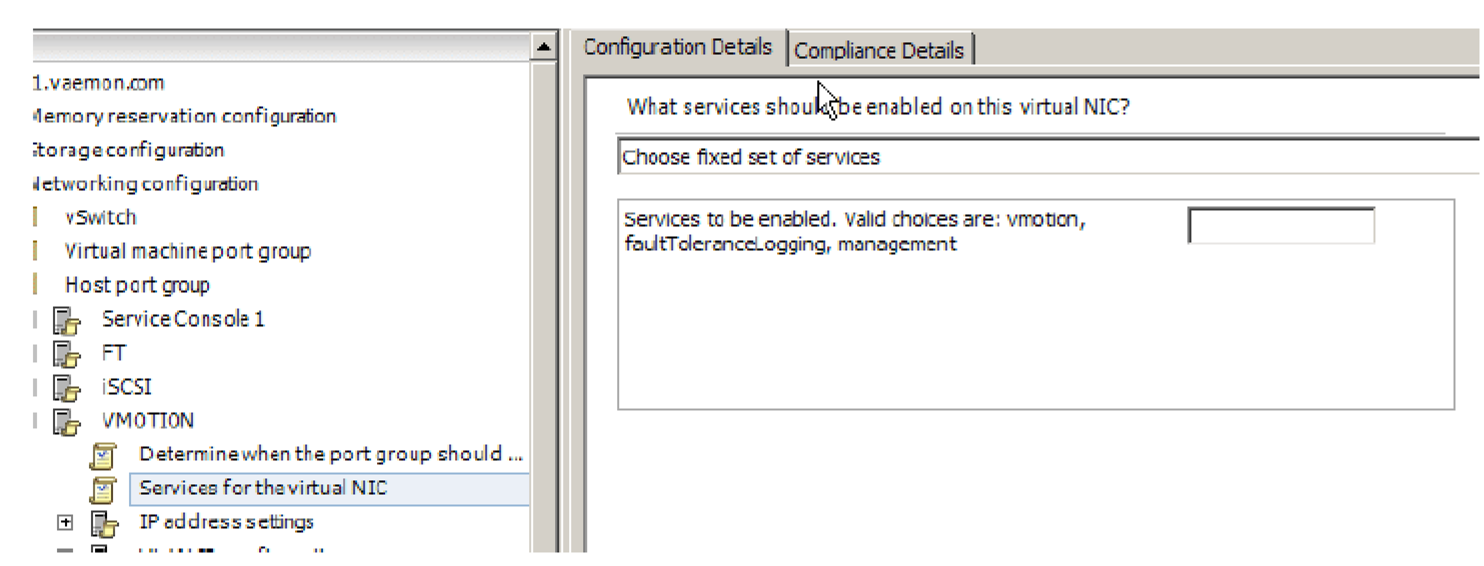
▶ 管理 vSphere 中的 Profile

1. 在 Profile 管理画面中，选择你要修改的选项。举例来说，我们希望 VMOTION 这个 VMkernel 能支持 VMotion 功能。先选择在 Host port group 中创建的 VMOTION。



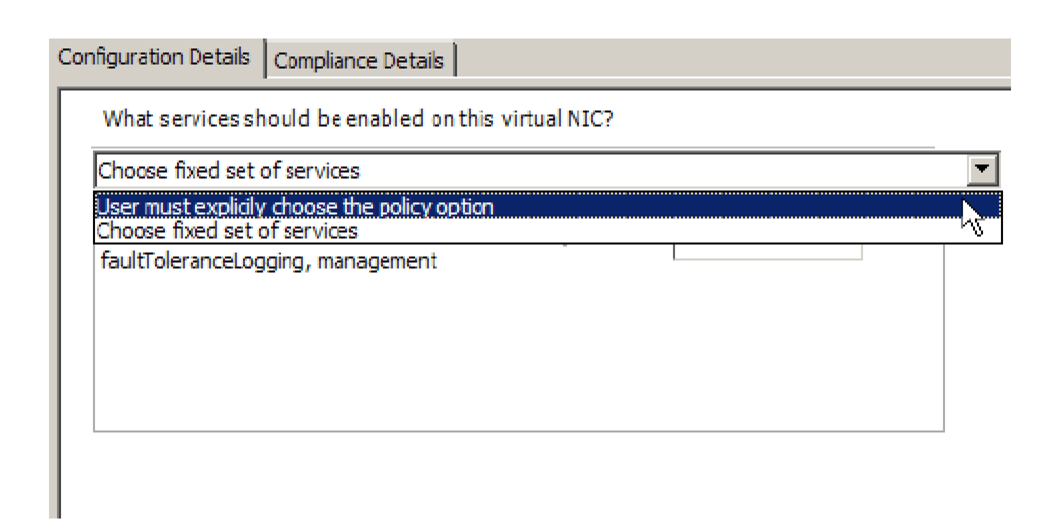
▲ 选择刚才创建的 VMOTION

2. 打开这个群组，并且选择 Services for the virtual NIC。在窗口右边会弹出要更改的值。



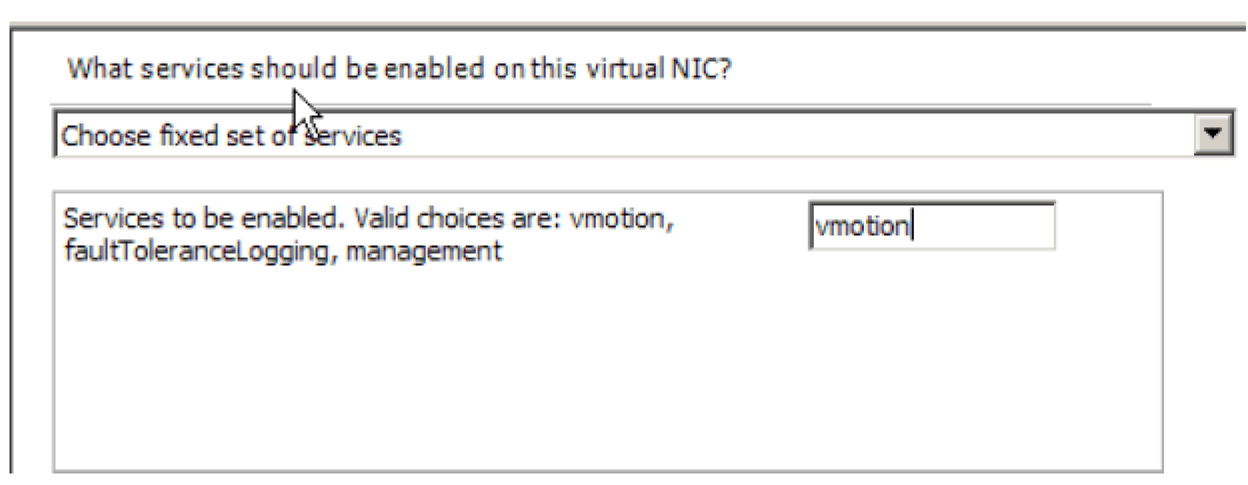
▲ 选择这个 NIC 的服务

3. 这个值有两个选项，一个是在套用到新主机时让用户自行选择，另一个则是让你先行键入。



▲ 可以在导入后自行选择，或是直接给一个值

4. 我们就在此键入小写的 vmotion，意味着这个 VMkernel 需要有 VMotion 激活。键入完之后单击 OK 按钮。



▲ 这里键入一个值，导出后就不用再改了

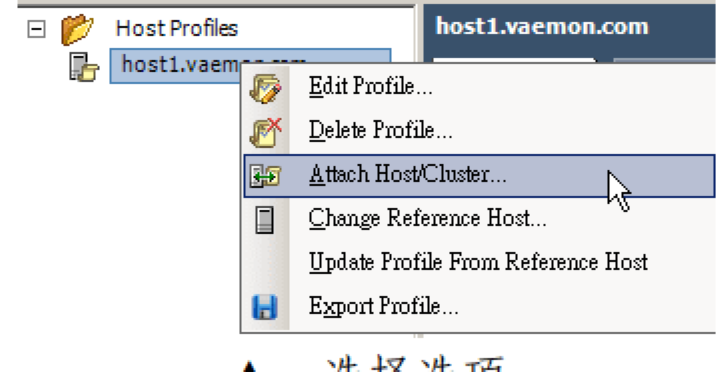
你也可以编辑其他的数值，以匹配具体部署的需求。

4. 将 Profile 连接上主机

当你要将一个 Profile 应用到新的主机上时，必须先将这个 Profile 连接到主机上，这个动作称为 Attach。在连接上之后，新的主机才能套用到这个 Profile。

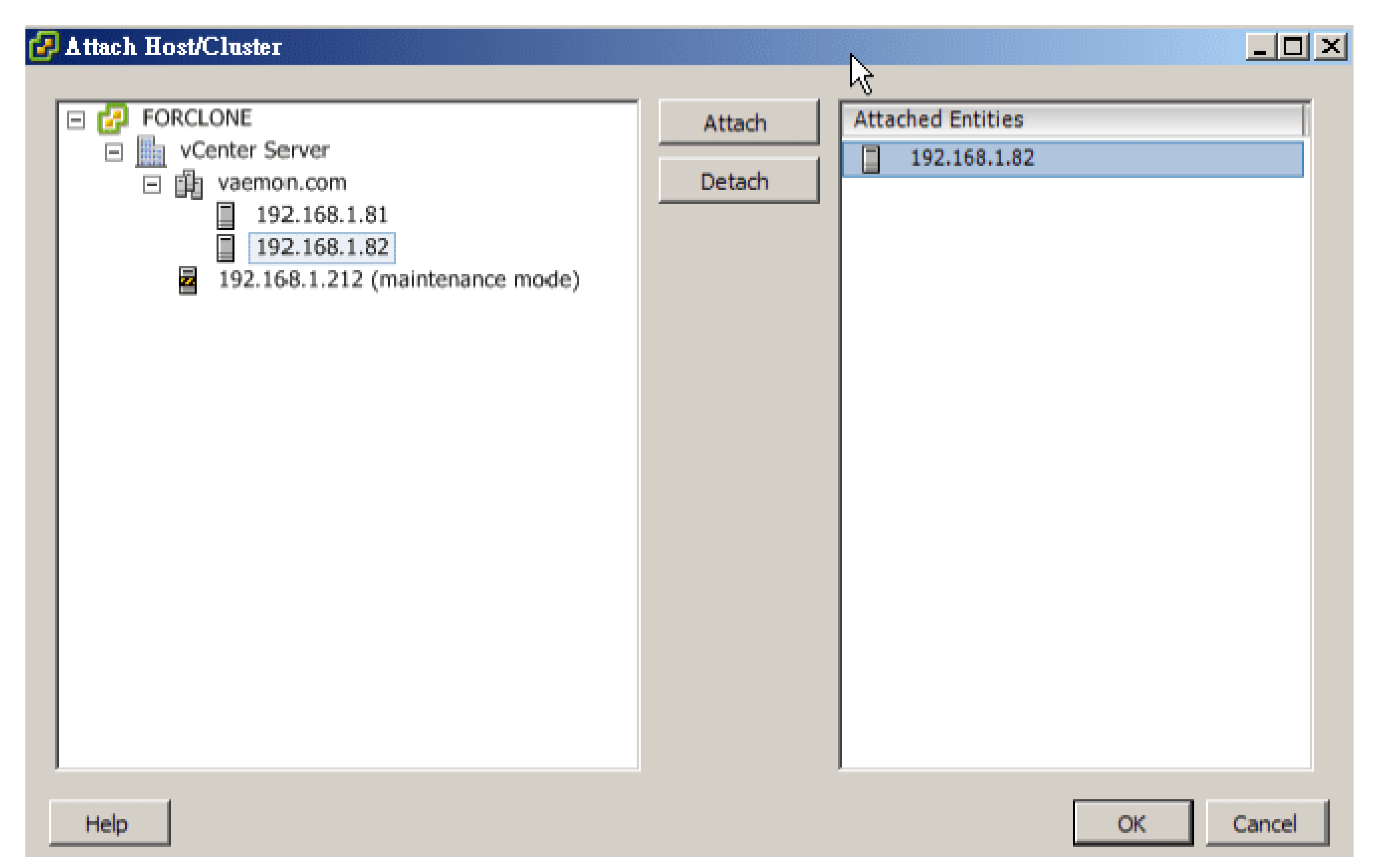
▶ 将 Profile 连接到新的主机上

1. 在 Profile 上右击，选择 Attach Host/Cluster 选项。



▲ 选择选项

2. 在画面中选择你要套用的主机，并且单击 Attach 按钮，就会到窗口的右边。这里可以选择多台主机。选完之后单击 OK 按钮。



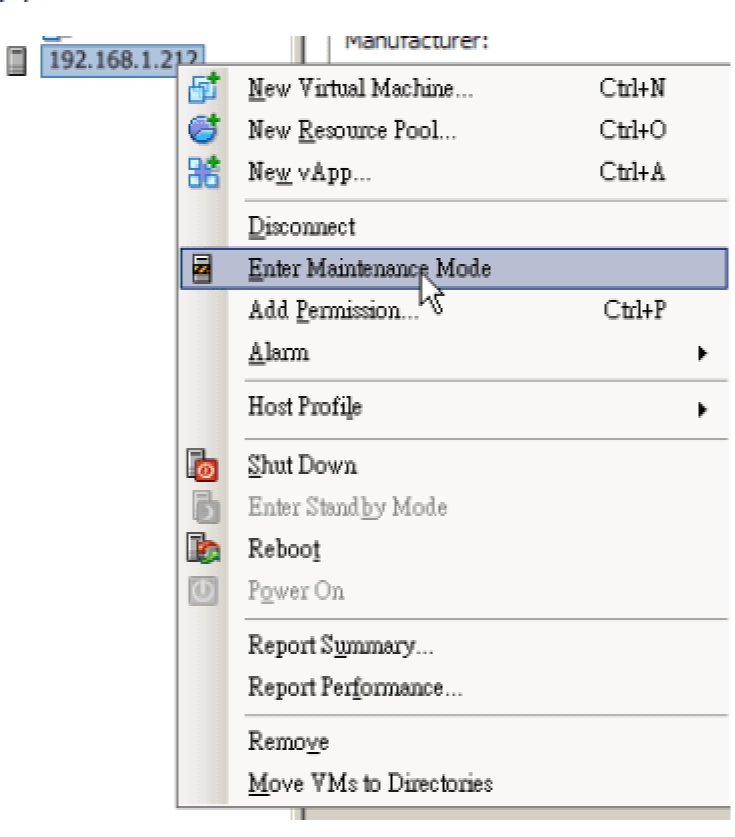
▲ 要先连接上主机才能进行导入

5. 将 Profile 套用到新主机上

将 Profile 连接之后，就可以开始套用了。由于许多细节还是需要重新键入，因此在套用过程中，vCenter 会将需要重新键入的地方让用户键入。

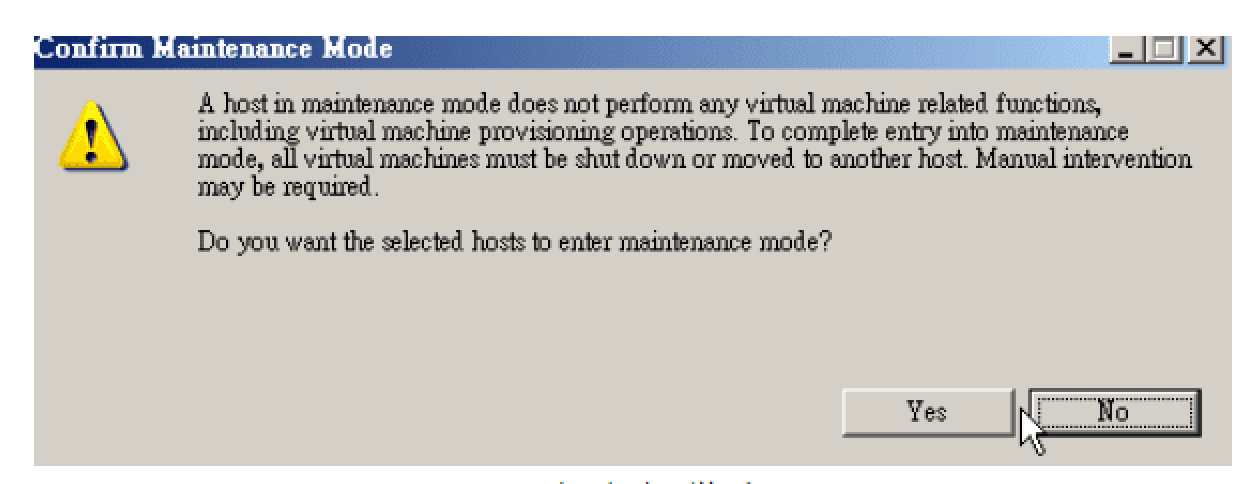
▶ 将 Profile 套用到新主机上

1. 首先必须将 ESX 主机进入维修模式（Maintenance mode）。在主机上右击，选择 Enter Maintenance Mode 选项。



▲ 要先进入维修模式

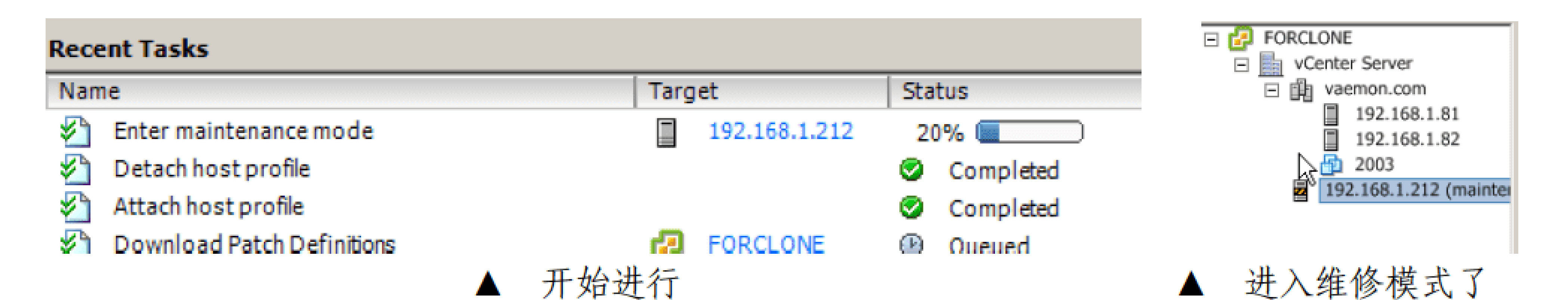
2. 此时会弹出报警窗口，单击 Yes 按钮即可。



▲ 会有报警窗口

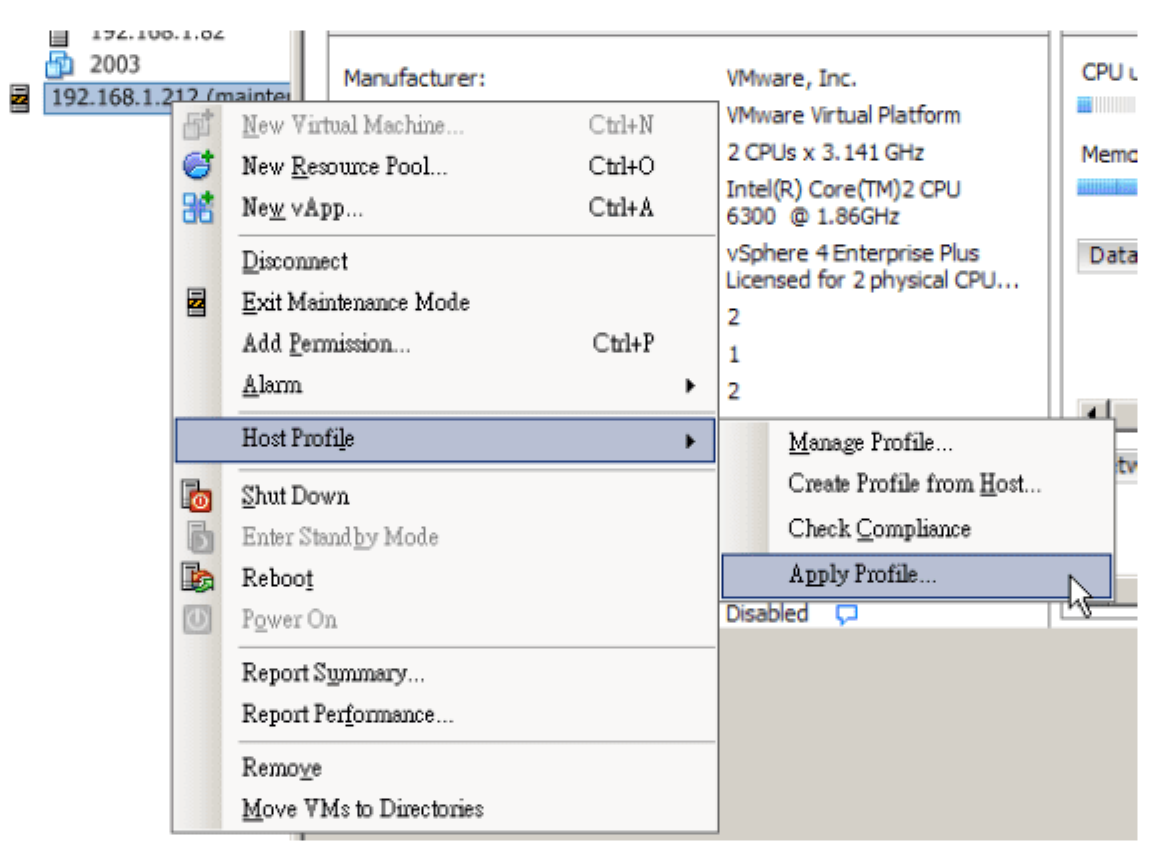
3. 开始进行。

4. 当主机弹出图中的图案时，意味着进入维修模式了。



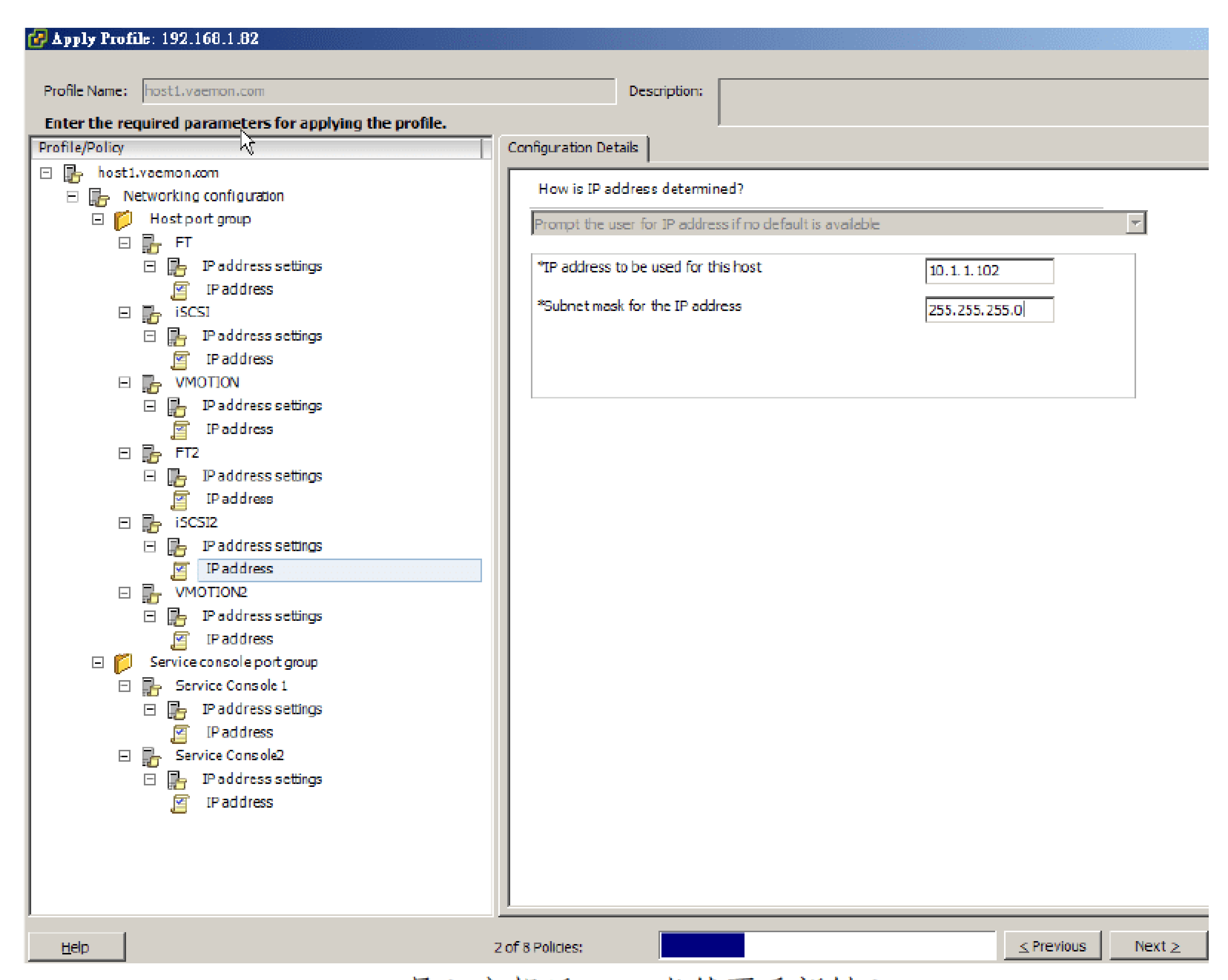
▲ 开始进行　　　　▲ 进入维修模式了

5. 在该主机上右击，选择 Host Profile/Apply Profile 选项。



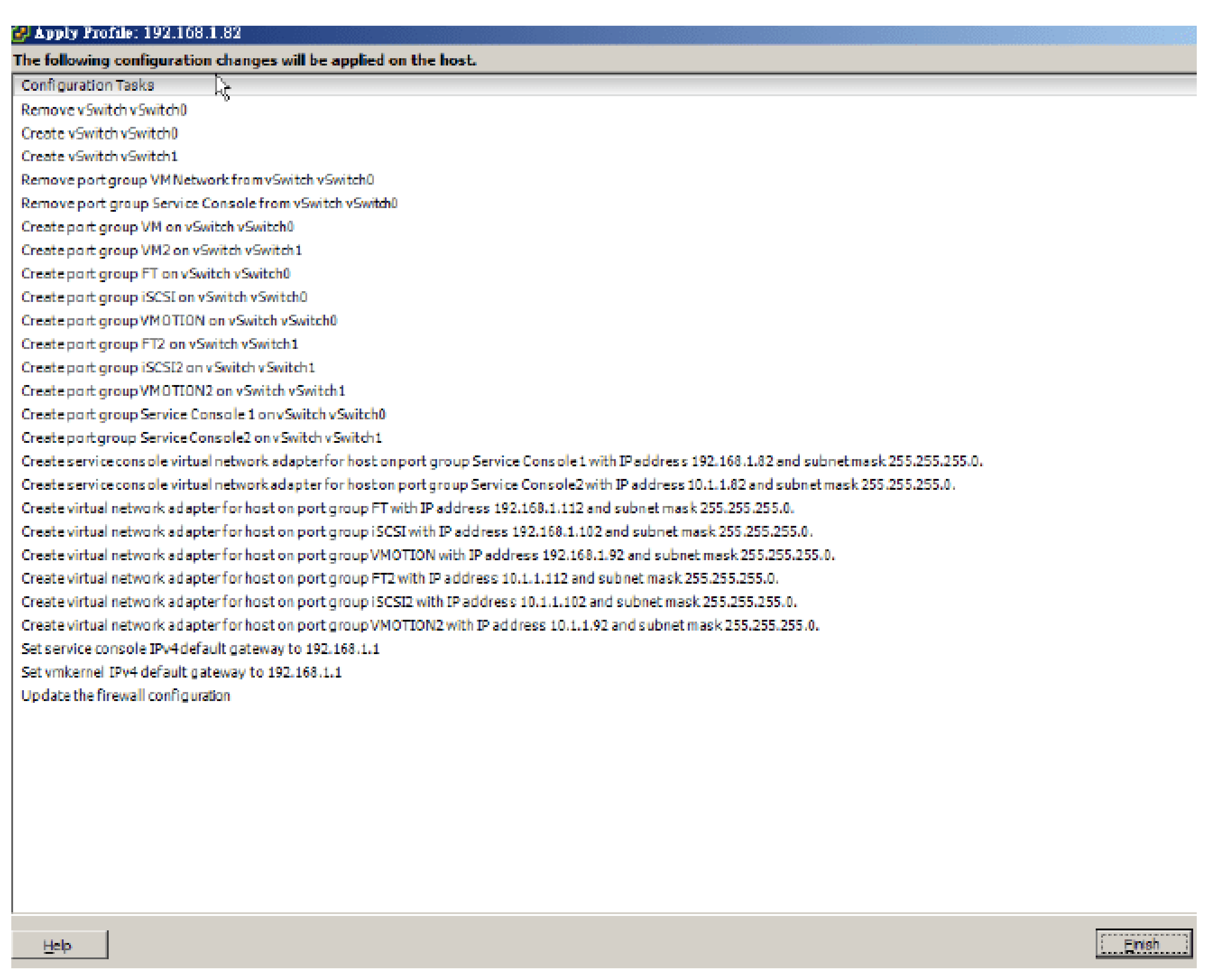
▲ 导入刚才连接的 Profile

6. 接下来会进入套用窗口。首先会要求键入几个 VMkernel 的 IP。要特别注意的是，这里的 IP 键入不会照顺序。从图中可以看到一共有 8 个键入项要键入。



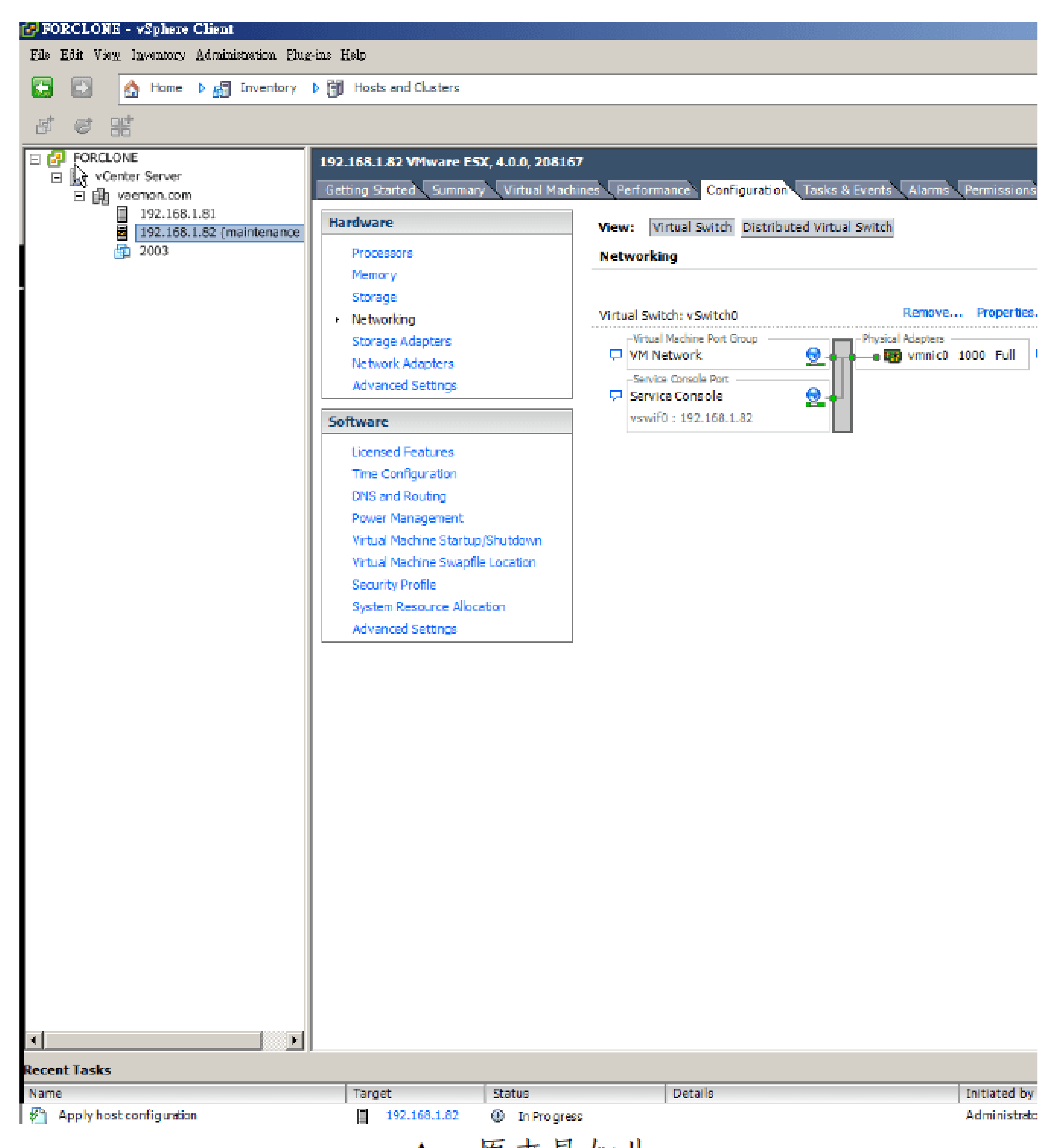
▲ 导入主机后，IP 当然要重新键入

7. 当 8 个键入项都键入落实之后，会有一个总结窗口，单击 Finish 按钮即可落实。



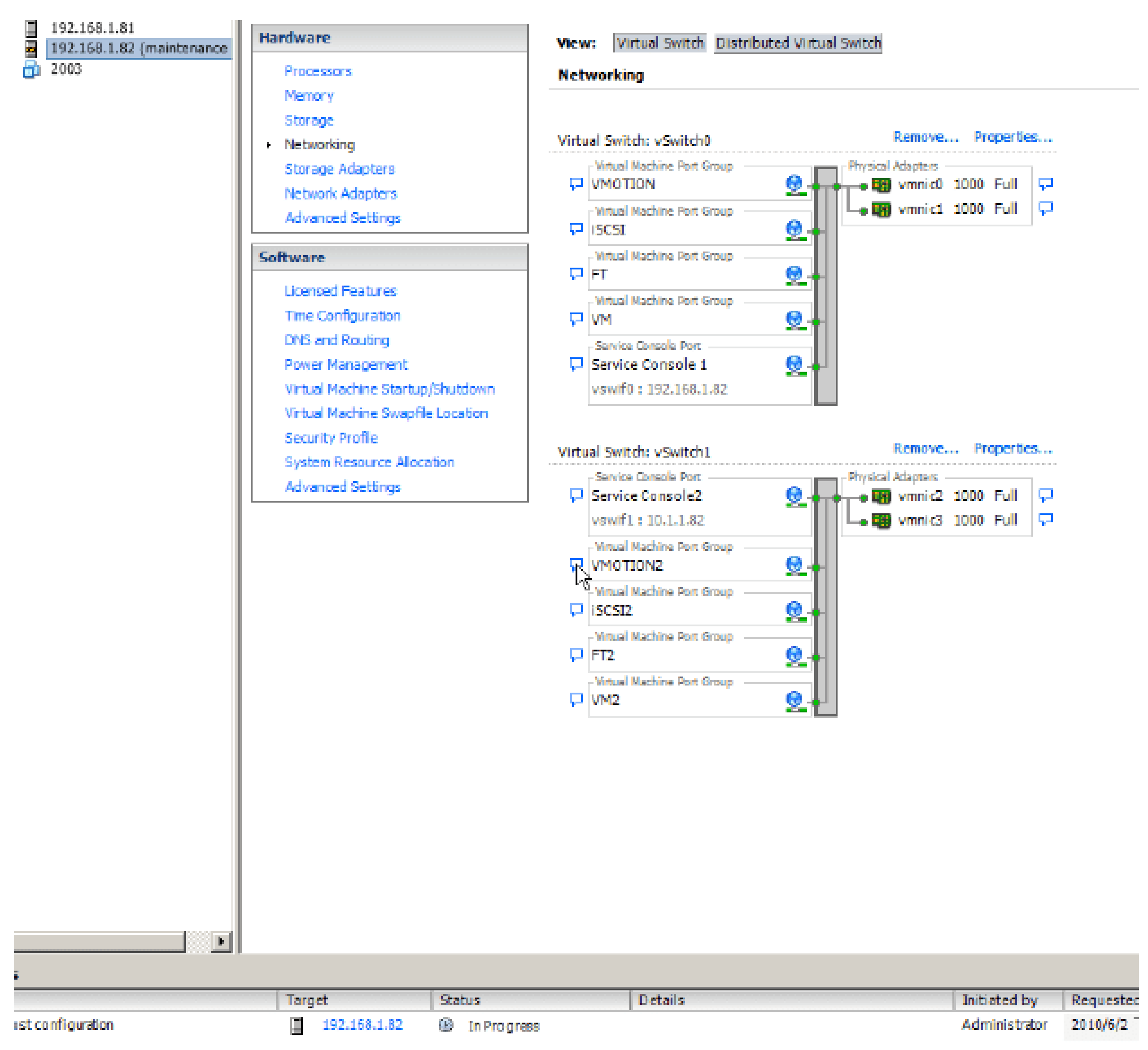
▲ 导入落实

8. 原来的网络配置是如此。



▲ 原来是如此

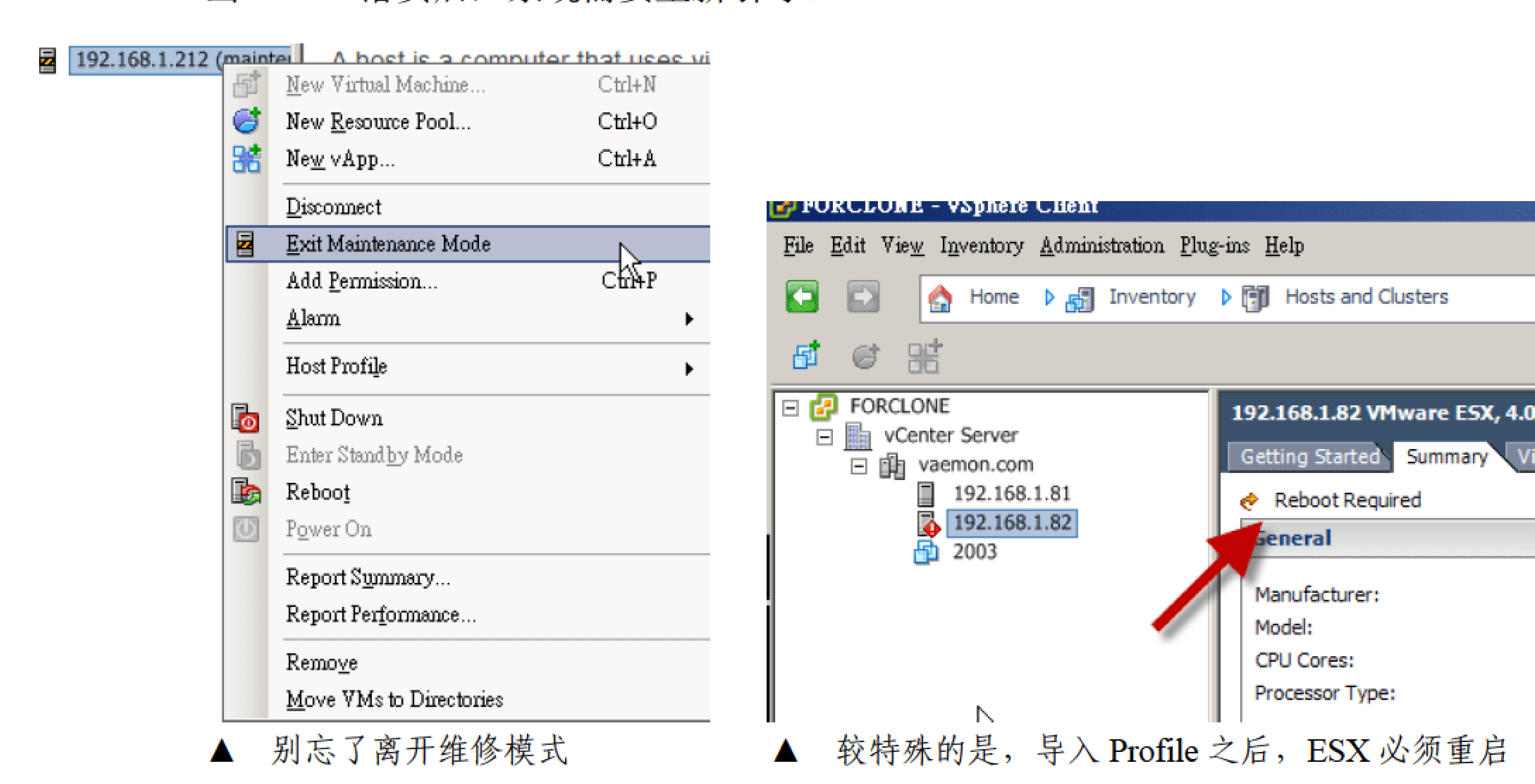
9. 当 Profile 导入之后，网络配置会自动变成和 host1 一样。



▲ 立即导入成功

10. 接下来不要忘了离开 Maintenance Mode。

11. 当 Profile 落实后，系统需要重新引导。



▲ 别忘了离开维修模式　　▲ 较特殊的是，导入 Profile 之后，ESX 必须重启

结 语

熟悉 vCenter 是玩转 vSphere 架构最重要的基础，而 ESX/ESXi 主机操作更是最重要的一环。从网络、存储，一直到整个 vCenter 的界面实践，都必须很熟练。本章使用了 VMware 最有名的 Workstation 产品来类比 vSphere 环境，让没有物理服务器的用户也能在自身的 PC 上创建一个 vSphere 环境，而 vCenter 的各项部件更是本章的热点，而 Profile 和 Template 的操作，将会让你在 vSphere 中省下大量的时间和精力。

读书笔记

第 15 章 所有高级功能的基础：VMotion 以及 Storage VMotion

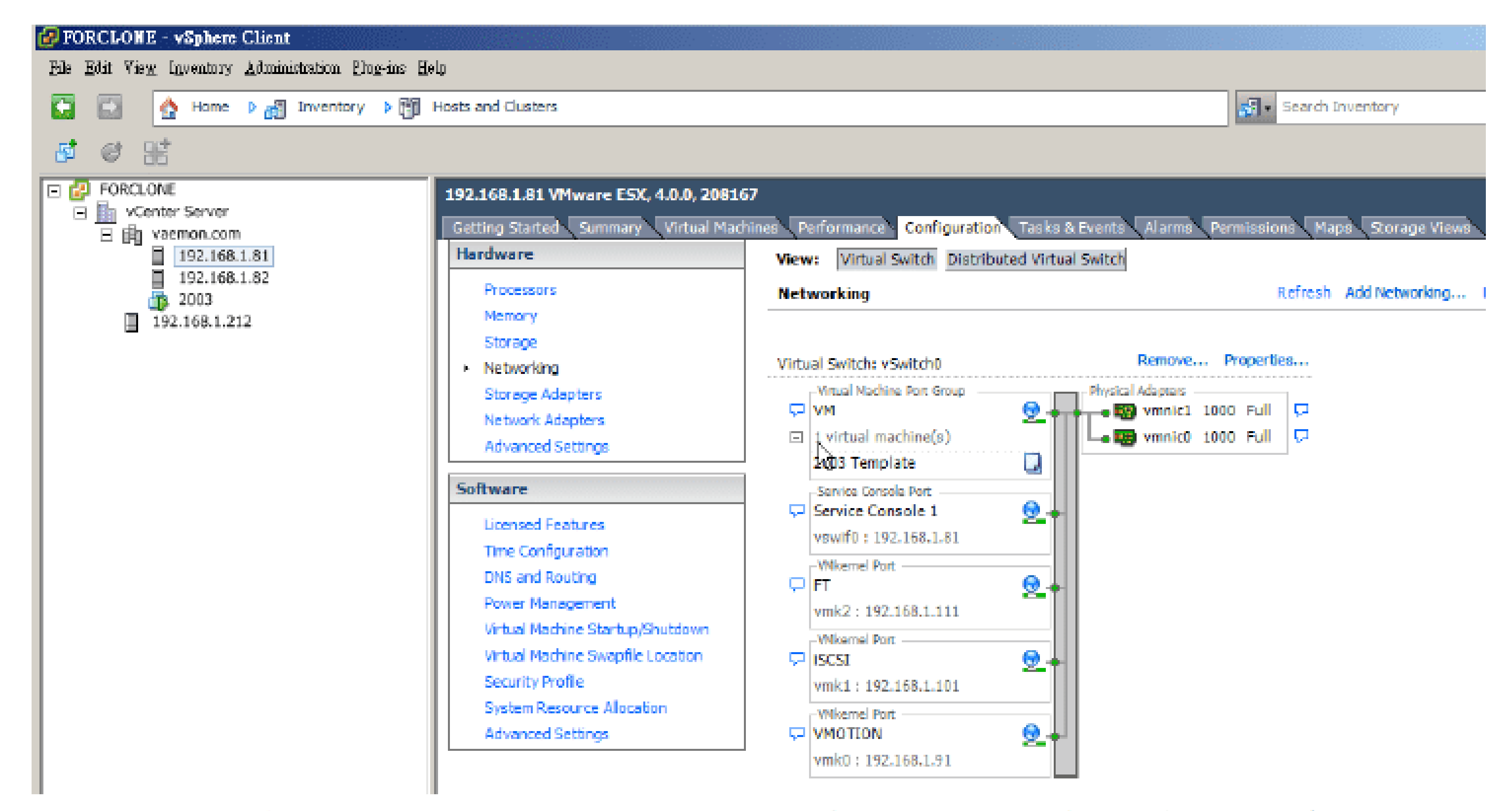
关键词：

- VMotion 是所有 vSphere 高级功能的重要基础
- 理解运行 VMotion 之前的网络配置
- 理解运行 VMotion 之前的存储配置
- 理解 VMotion 的原理
- 理解什么是 EVC 模式
- 开始进行非引导状态的 VMotion
- 进行引导状态的 VMotion
- 理解 SVMotion 的原理
- 进行非引导状态的 SVMotion
- 运行引导状态的 SVMotion

VMware 是全世界最大的虚拟机厂家，其产品占据了 75%以上的市场份额，其产品线的完整、功能的强大、配置的复杂、需要硬件的配合程度，都一再显示了其整体解决专案的多面性。VMware 钻研企业级的应用有十多年的经验，当竞争对手还在埋头单个的虚拟机产品时，VMware 已经开发出以虚拟机技术为主的整体解决专案了。VMware 的 VI 产品线，不是一个虚拟机产品，而是一个“以虚拟机为技术”的企业整体解决专案。专案中的整体服务，可以让系统蓝屏、计划停机等恶梦完全消失，提供企业永远不中断的服务。我们在这一章中，就来看看 vSphere 中最值得期待的技术——VMotion。

15.1 使用 VMotion 环境的准备任务

承续上一章的生产环境，我们在三台物理服务器上安装了 vCenter Server 及 ESX 的环境，也使用了一台 VMware Workstation 安装了测试环境，在这一章的教程中，将会分别以这两个环境来实践 VMotion。为了更有效类比具体的环境，我们必须先进行整个网络以及服务器的规划，才能运行 VMotion 等功能。



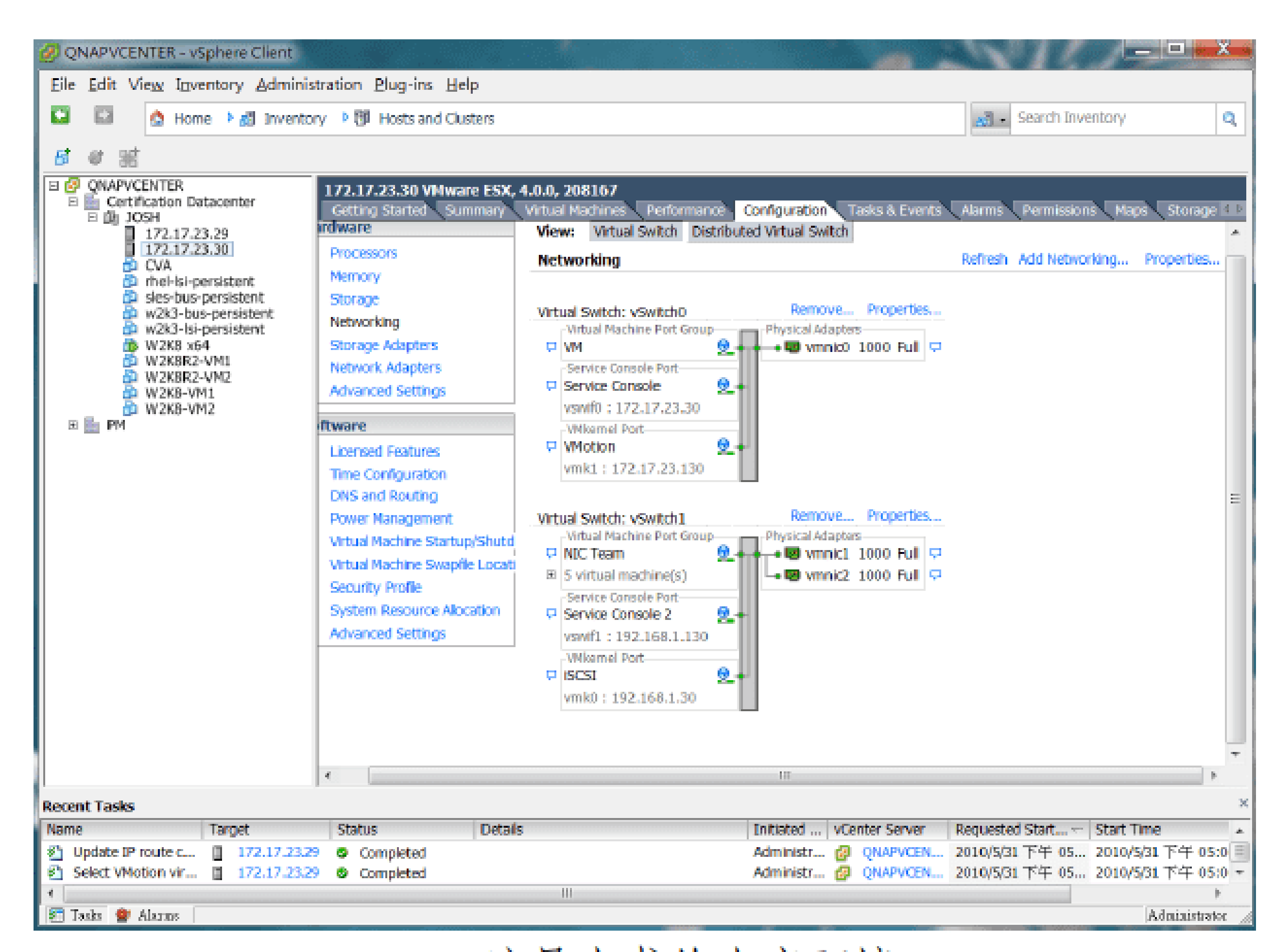
▲　网络环境是 VMotion 的重要因素，因为要靠网络来搬动 VM

15.1.1　网络的划分

在一个完整的 vSphere 环境中，不同的功能走不同的网络路由是十分重要的。在本章 vSphere 的生产环境中使用两个网段，分别是 192.168.1.X 以及 172.17.23.X。其中 192.168.1.X 使用了两片网卡做 NIC Team，作为 VMotion 以及 iSCSI 的连接，而 172.17.23.X 则使用一片网卡来做 Service Console。

▶ **本章物理生产环境的网络分配**

- 第一片网卡：172.17.23.X/255.255.254.0
- 第二/三片网卡：192.168.1.X/255.255.255.0

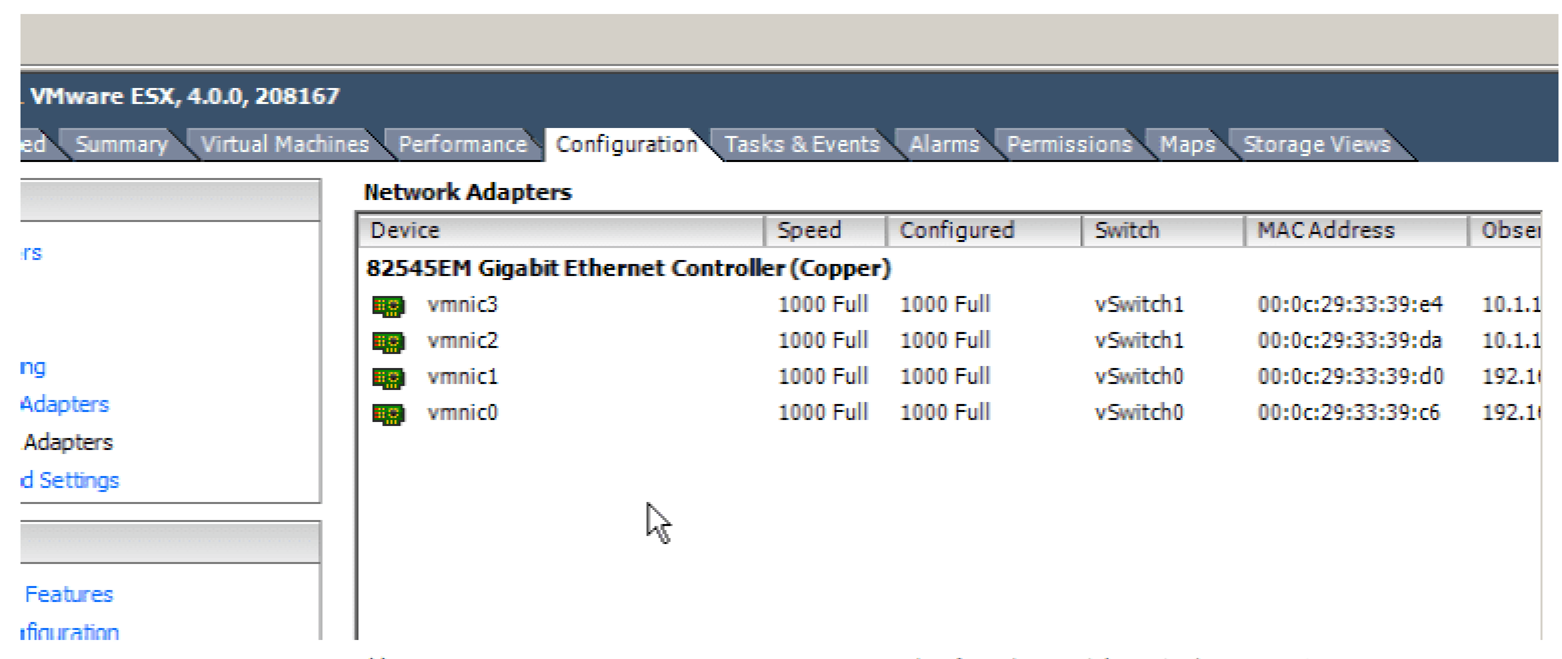


▲　这是本书的生产环境

由于生产环境需要较多计算机，因此一个 C 级网段的 IP 可能不够，因此我们开放 172.17.22.X 也可以使用到这个网段，如此一来，就有较多的计算机可以加入这个网段，不管是虚拟或是物理机都一样。

▶ **本章测试环境的网络分配**

- 第一/二片网卡：192.168.1.X/255.255.255.0
- 第二/三片网卡：10.1.1.X/255.255.255.0



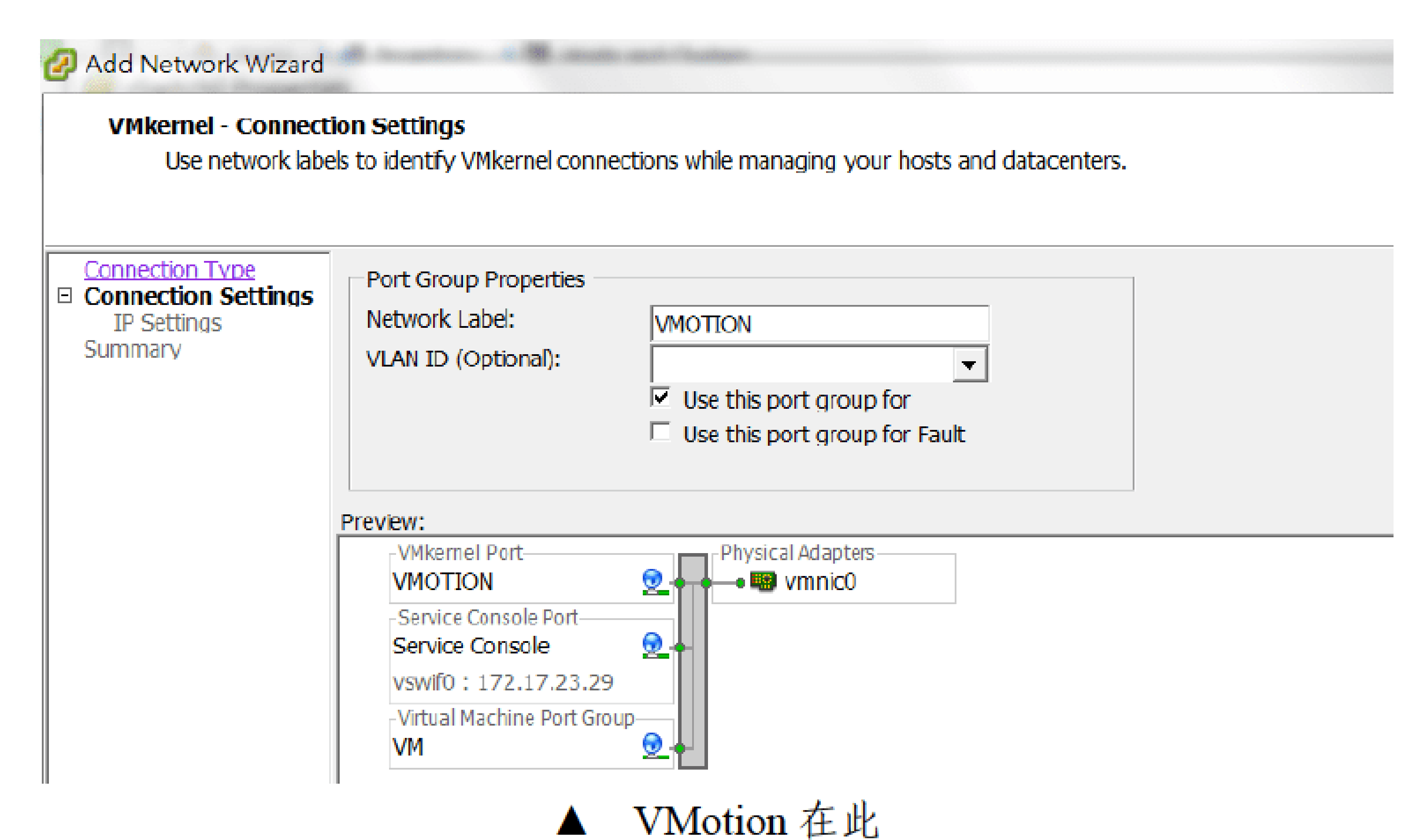
▲ 使用 VMware Workstation 的实验环境网卡配置

15.1.2 生产环境虚拟交换机的划分

在本章的示例中，我们分别在三张不同的网卡上创建两个虚拟交换机 vSwitch。其上并且有不同的功能。由于 ESX 的网络功能都必须依赖虚拟交换机，因此我们会在创建功能时并发创建虚拟交换机，并且绑定到不同的网卡上。至于实验环境的虚拟交换机，则参阅上一章的网络划分部分，在此不再重复。

1. vSwitch0

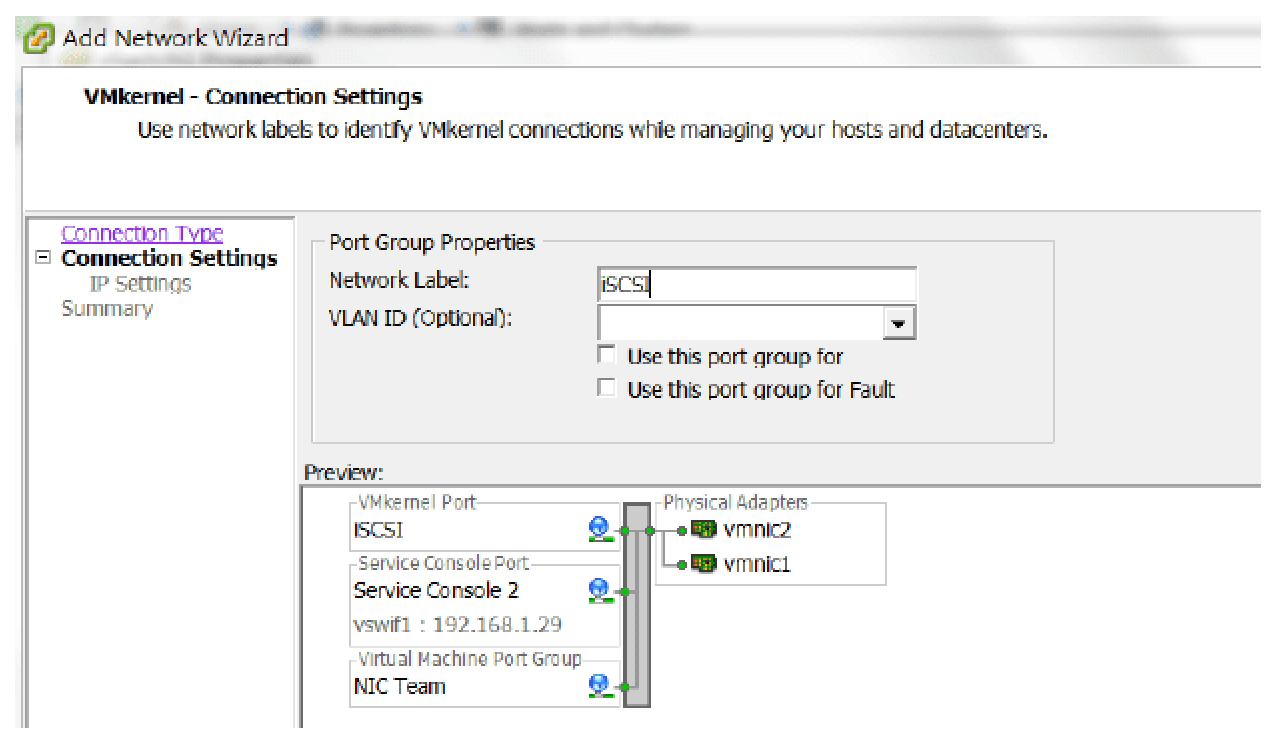
这个交换机绑定在 vmnic0 上，是系统默认的交换机。上面有默认的 Service Console 和给虚拟机使用的 Virtual Machines 通信端口组，当然还有 VMotion 的信道。



▲ VMotion 在此

2. vSwitch1

这个交换机绑定在 vmnic1/2 两张网卡上，主要是以给 VM 用的通信端口组、第二个副本的 Service Console 以及存储的 iSCSI。



▲ iSCSI 在此

15.1.3 ESX 通信端口/通信端口组功能划分

当网卡、虚拟交换机都分配好之后，我们就必须在每一台 ESX 上配置不同的通信端口/通信端口组功能了。下面就来看看每一个通信端口/通信端口组的功能。

1. Service Console

这个位于 vmnic0 的 vSwitch0 上的主控台，是安装 ESX 时默认的。主要是给 vCenter Server 管理用。使用的 IP 为 172.17.23.29。

2. Service Console 2

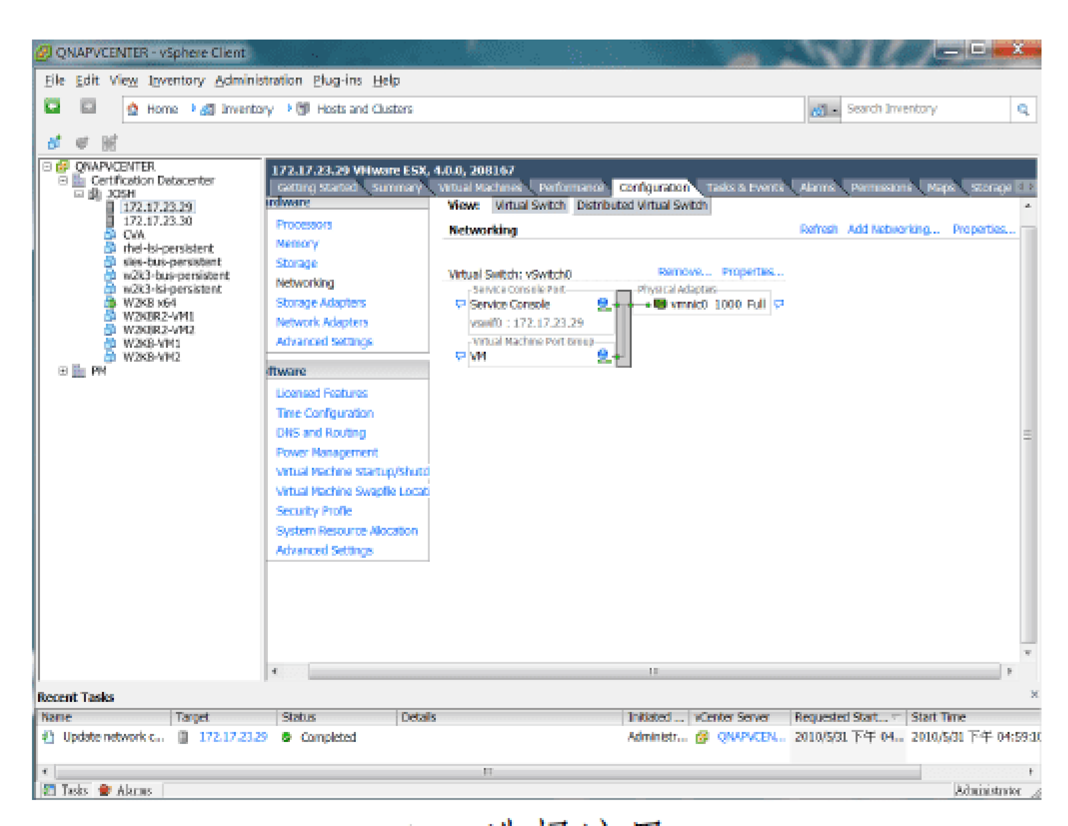
这是位于 vmnic1/2、vSwitch2 上的主控台，我们必须手动安装。安装的目的是为了测试同一网卡上，同一虚拟交换机上的 iSCSI VMkernel 是否能连通。使用的 IP 为 192.168.1.29。

3. VMKernel VMOTION

这是位于 vmnic0、vSwitch0 上的 VMkernel 通信端口。必须手动安装，安装时必须确定 VMotion 激活。使用的 IP 为 172.17.23.129。在理解运行 VMotion 的网络配置之后，接下来我们就以上面的流程来制作一个 VMotion 专用的 VMkernel 通信端口，替读者们复习一下制作 VMkernel 的方式。

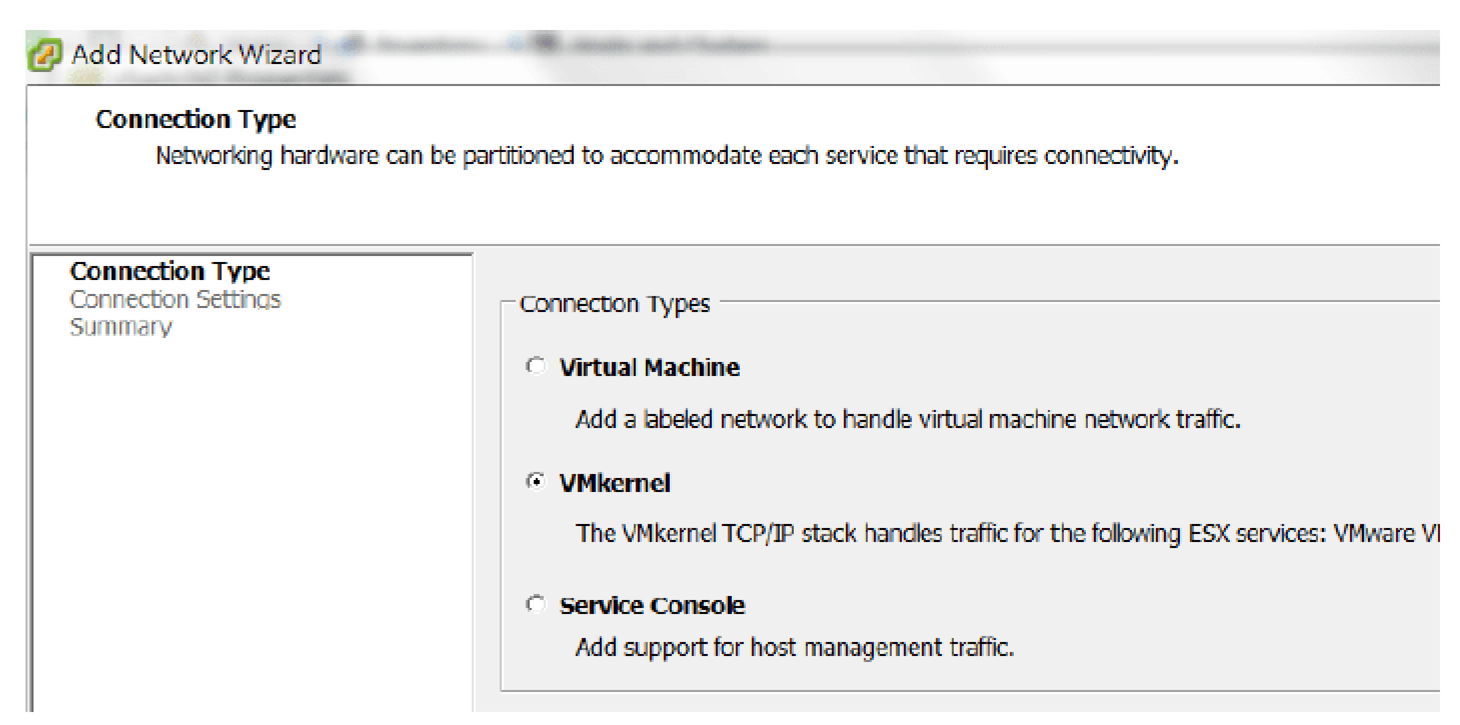
▶ 制作 VMotion 专用的 VMkernel 通信端口

1. 进入 vCenter Server，并且在 172.17.23.29 这台 ESX 主机上，选择 Configuration/Networking/Add Networking。



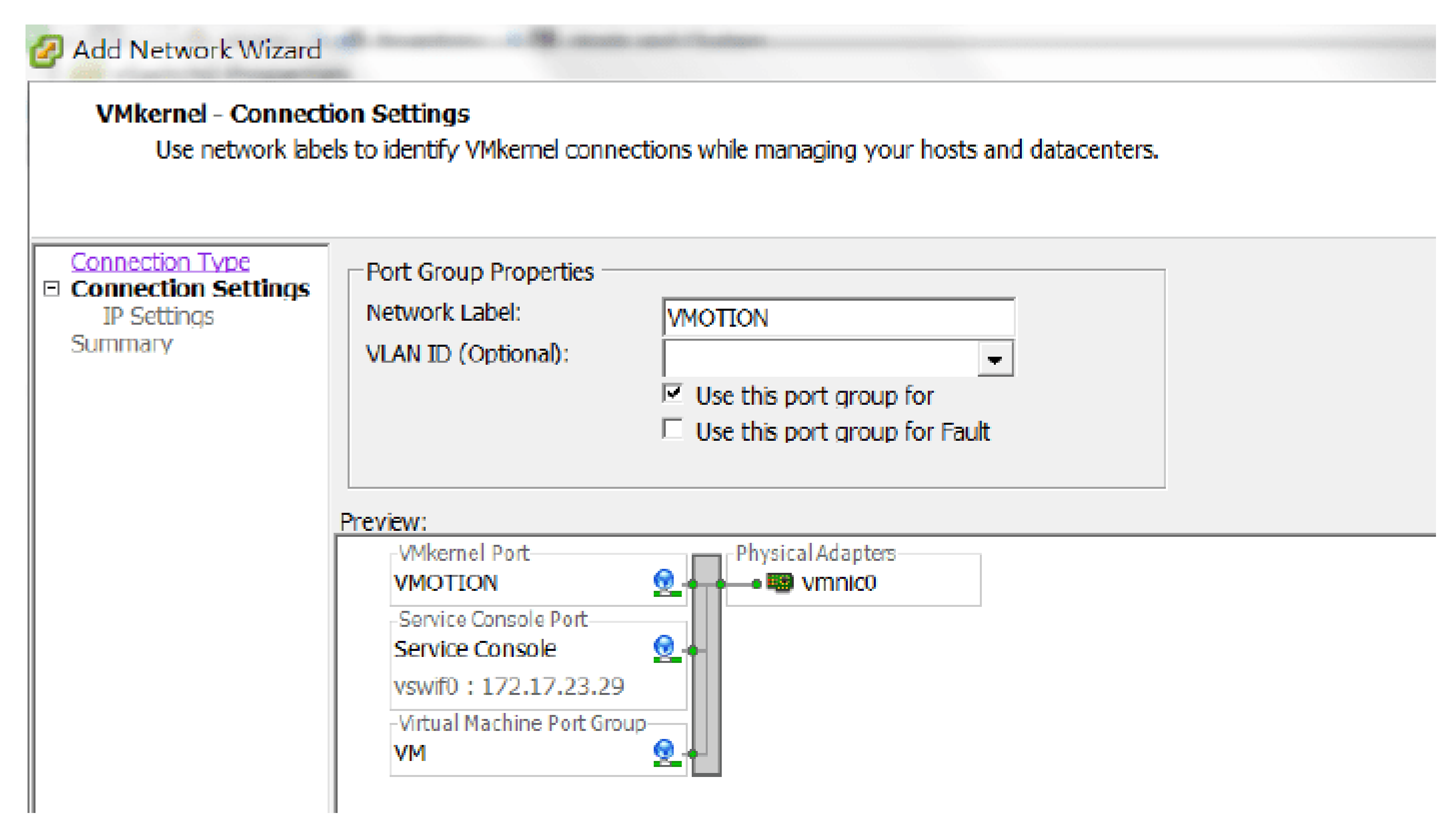
▲ 选择这里

2．由于 VMotion 的动作必须使用 VMkernel，因此我们选择 VMkernel。



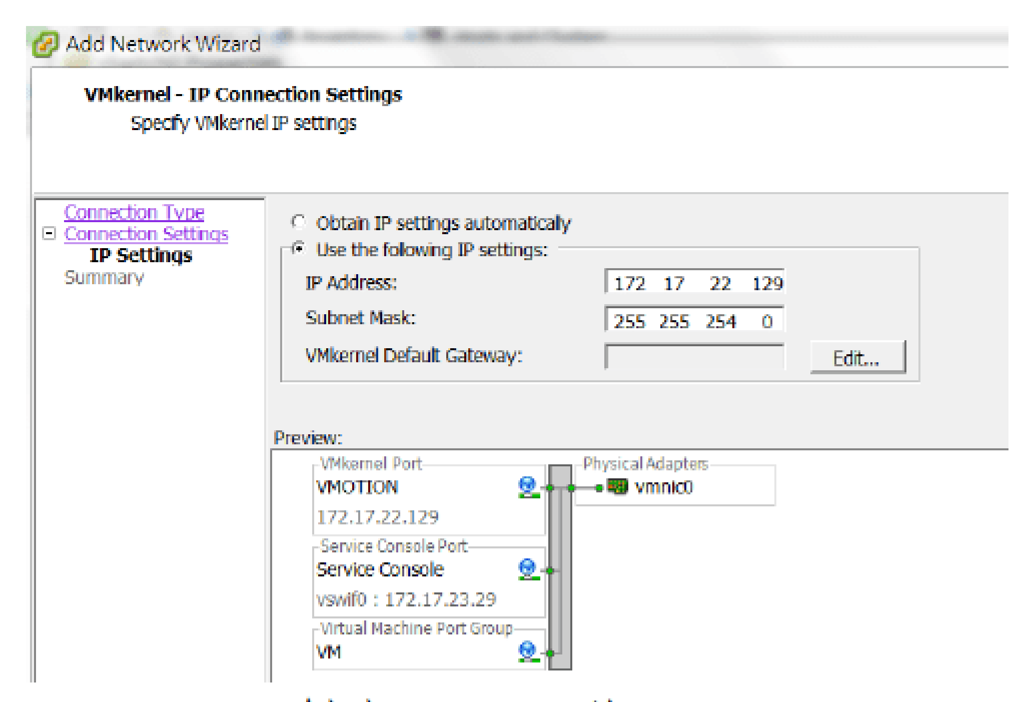
▲ 选择第二项

3．使用原来的网卡 vmnic1，并且单击 Next 按钮。



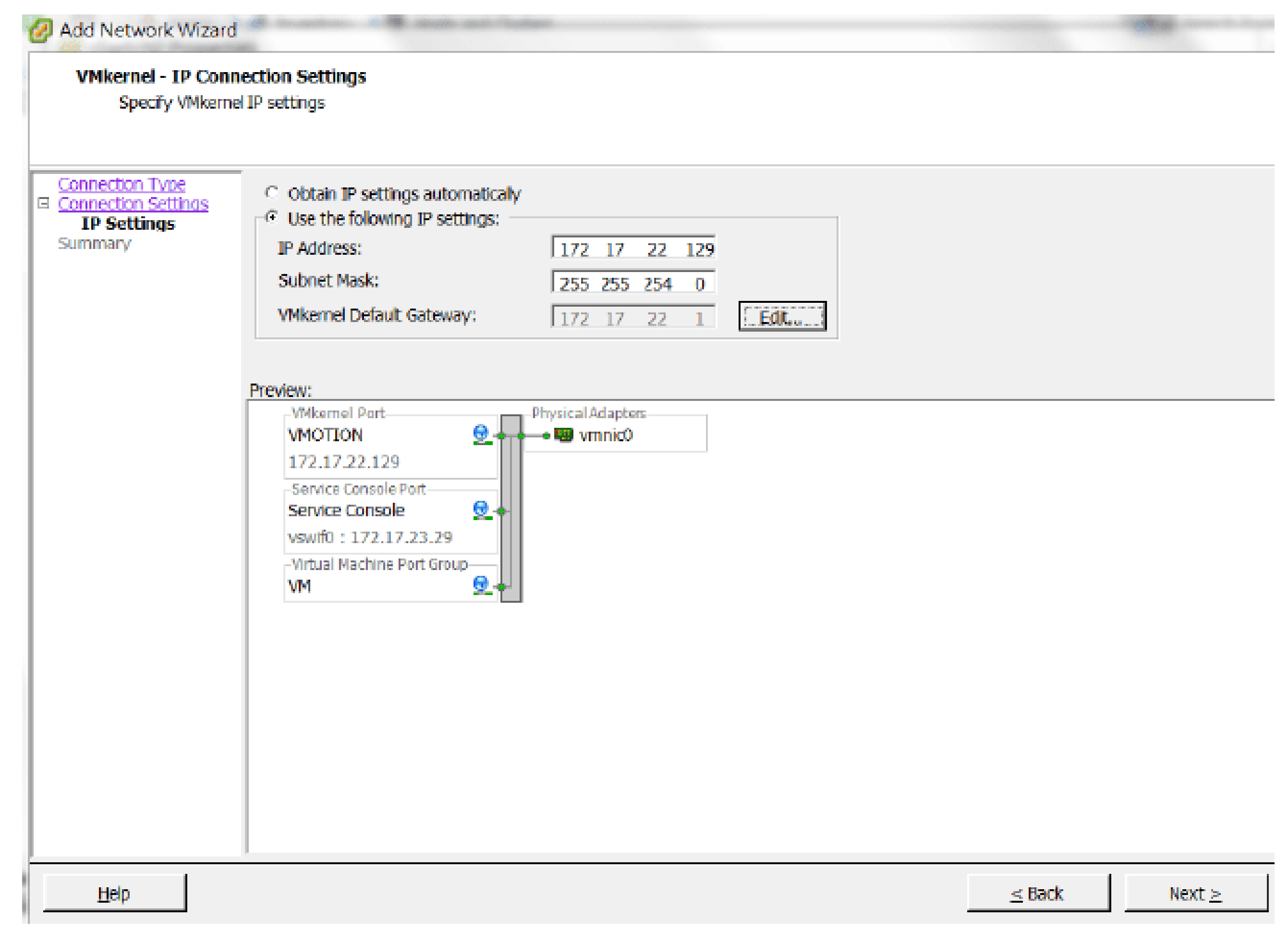
▲ 使用原来的网卡来连接

4．在 Network Label 的列表框键入大写的 VMOTION，最重要的就是勾选 Use this port for VMotion 复选框。接下来键入这个通信端口的 IP（172.17.23.129）和 Subnet mask（255.255.254.0）。使用 VMotion 的通信端口需要 Gateway，这个网段的 Gateway 是 172.17.22.1。



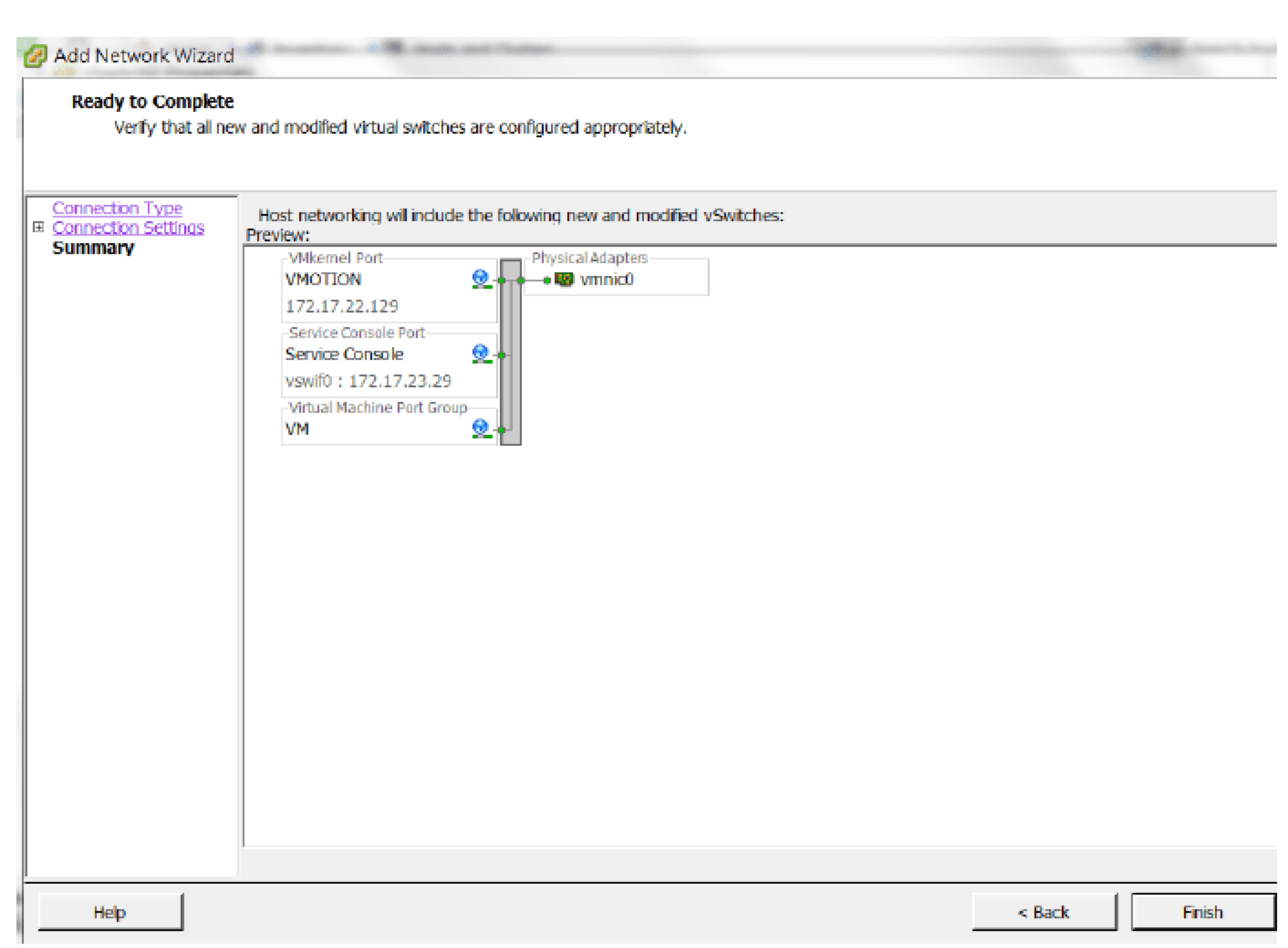
▲ 创建 VMware 的 VMkernel

5．此时会弹出落实之后的网络示意图，单击 Finish 按钮继续。



▲ 总结画面

6．之后我们可以看到这个 VMkernel 已经创建落实，专门给 VMotion 用。



▲ 创建落实

4. VMkernel：iSCSI

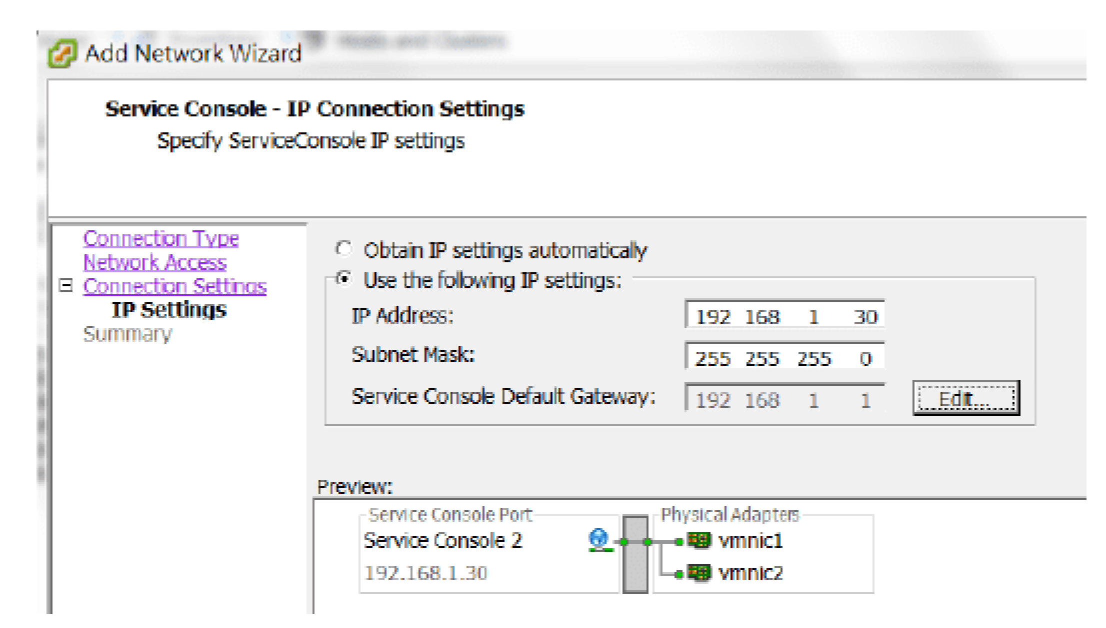
这是位于 vmnic1/2、vSwitch1 上的 VMkernel 通信端口，必须手动安装，主要是用于 iSCSI 的通信。使用的 IP 为 192.168.1.29。

5. Virtual Machines：VM

这是位于 vmnic0、vSwich0 上的 Virtual Machines 通信端口组。必须手动安装，主要是用于将 ESX 上虚拟机和物理网络段相连接的通信端口组，使用的 IP 为 172.17.23.x。

6. Virtual Machines：NIC Team

这是位于 vmnic1/2 上，vSwitch1 上的 Virtual Machines 通信端口组。必须手动安装，主要是用于将 ESX 上虚拟机和物理网络段相连接的通信端口组，使用的 IP 为 192.168.1.X。



▲ 这里特别使用了 NIC Team 来做多网卡的冗余绑定

7. 本章 vSphere 示例的网络总汇总

有关新建虚拟交换机、通信端口组的详细步骤，读者们可以引用第 18 章中的完整说明，我们在此就不再重复了。而在将所有的网卡、虚拟路由器和通信端口/通信端口组配置完之后，我们特别将这些网络配置做一个总汇总，在配置时可以参阅表 1：

表 1 vSphere 中的网络路由组合

	网卡#0	网卡#1	网卡#2
网段	172.17.23.29/30 255.255.254.0 172.17.22.1	192.168.1.29/30 255.255.255.0 192.168.1.1	
在 ESX 上的名称	vmnic0	vmnic1	vmnic2
所在虚拟交换机	vSwitch0	vSwitch1	
Service Console	有（系统默认） 172.17.23.29/30	192.168.1.29/30	
VMkernel	VMOTION 172.17.23.129/130	iSCSI 192.168.1.129/130	
Virtual Machine Group	VM 172.17.23.X 网段	NIC Team 192.168.1.X	

15.1.4 vSphere 主机的配置及说明

下面是生产环境的真实情况，参加的设备如下。

▶ 参加 vSphere 生产环境的设备，以 IP 分

- 172.17.23.203：IBM X3550 M2 主机，安装 Windows 2003 Server，运行 vCenter
- 172.17.23.29：Dell R710 主机，安装 ESX
- 172.17.23.30：HP DL380 主机，安装 ESX
- 192.168.1.24/27：QNAP
- 网络设备：CISCO 交换机，划分 VLAN，分别是 172.17.23.X 以及 192.168.1.X

1. 172.17.23.203 主机

安装 Windows Server 2003，主要的功能 vCenter Server。

2. 192.168.1.24/27 QNAP 网络存储设备

NAS 主机，QNAP TS-859 Turbo，主要的功能为提供 iSCSI Target Server 及 NFS。

3. 172.17.23.29/30 ESX 主机

两台运行 ESX 的物理机，被 vCenter 主机管理。

4. 参与 vSphere 架构的主机总汇总

在本小节的最后，我们就来看看这四台参与 vSphere 架构的机器列表，读者们在安装前，可以和上面的网络列表做交叉汇总。

表 2　vSphere 的主机总汇总

主机名	172.17.23.203	192.168.1.24/27	172.17.23.29	172.17.23.30
位置	物理主机 IBM X3550 M2	NAS 网络存储 QNAP TS-859 Turbo	物理主机 DELL R710	物理主机 HP DL380 G6
功能	vCenter	iSCSI，NAS	ESX 服务器	ESX 服务器

5. 参与 vSphere 架构的实验环境汇总

上一章已经有完整的实验环境的 vSphere 架构，我们在此不详述其细节，仅给出其总汇总。

表 3　实验环境的 vSphere 架构

主机名	192.168.1	192.168.1.24/27	172.17.23.29	172.17.23.30
位置	物理主机 IBM X3550 M2	NAS 网络存储 QNAP TS-859 Turbo	物理主机 DELL R710	物理主机 HP DL380 G6
功能	vCenter	iSCSI，NAS	ESX 服务器	ESX 服务器

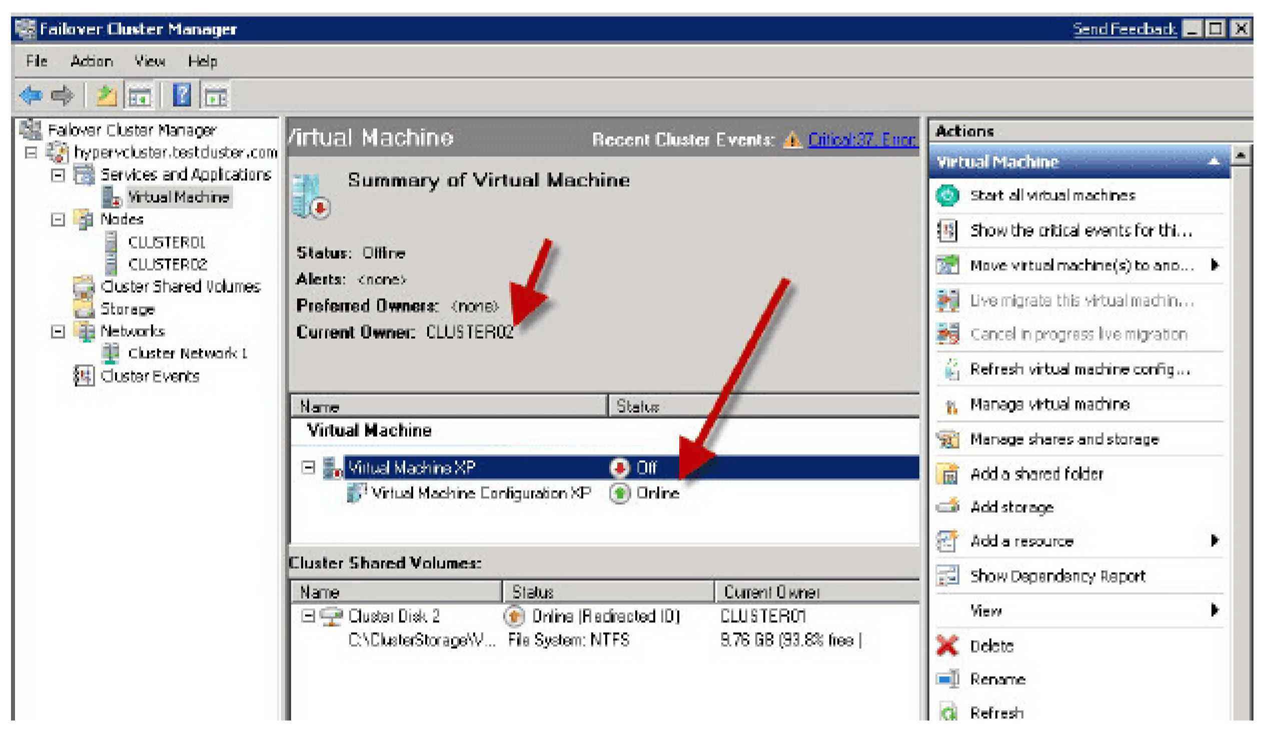
表 4　实验环境的 ESX 网络配置

	网卡#0	网卡#1	网卡#2	网卡#3
在 ESX 上名称	vmnic0	vmnic1	vmnic2	vmnic3
网段	192.168.1.81/82 255.255.255.0 192.168.1.1		10.1.1.181 255.255.255.0 192.168.1.1	
所在虚拟交换机	vSwitch0		vSwitch1	
Service Console	Service Console 1 192.168.1.81/82		Service Console 2 10.1.1.81/82	
Virtual Machine Group	VM 192.168.1.X 网段		VM2 10.1.1.X	
VMkernel	VMOTION 192.168.1.91/92 选中 VMotion 选项		VMOTION2 10.1.1.91/92 选中 VMotion 选项	

	网卡#0	网卡#1	网卡#2	网卡#3
VMkernel	iSCSI 192.168.1.101/102 所有选项不选中		iSCSI2 10.1.1.101/102 所有选项不选中	
VMkernel	FT 192.168.1.111/112 选中 FT Logging 选项		FT2 10.1.1.111/112 选中 FT Logging 选项	

15.2 vSphere 的动态转移——VMotion

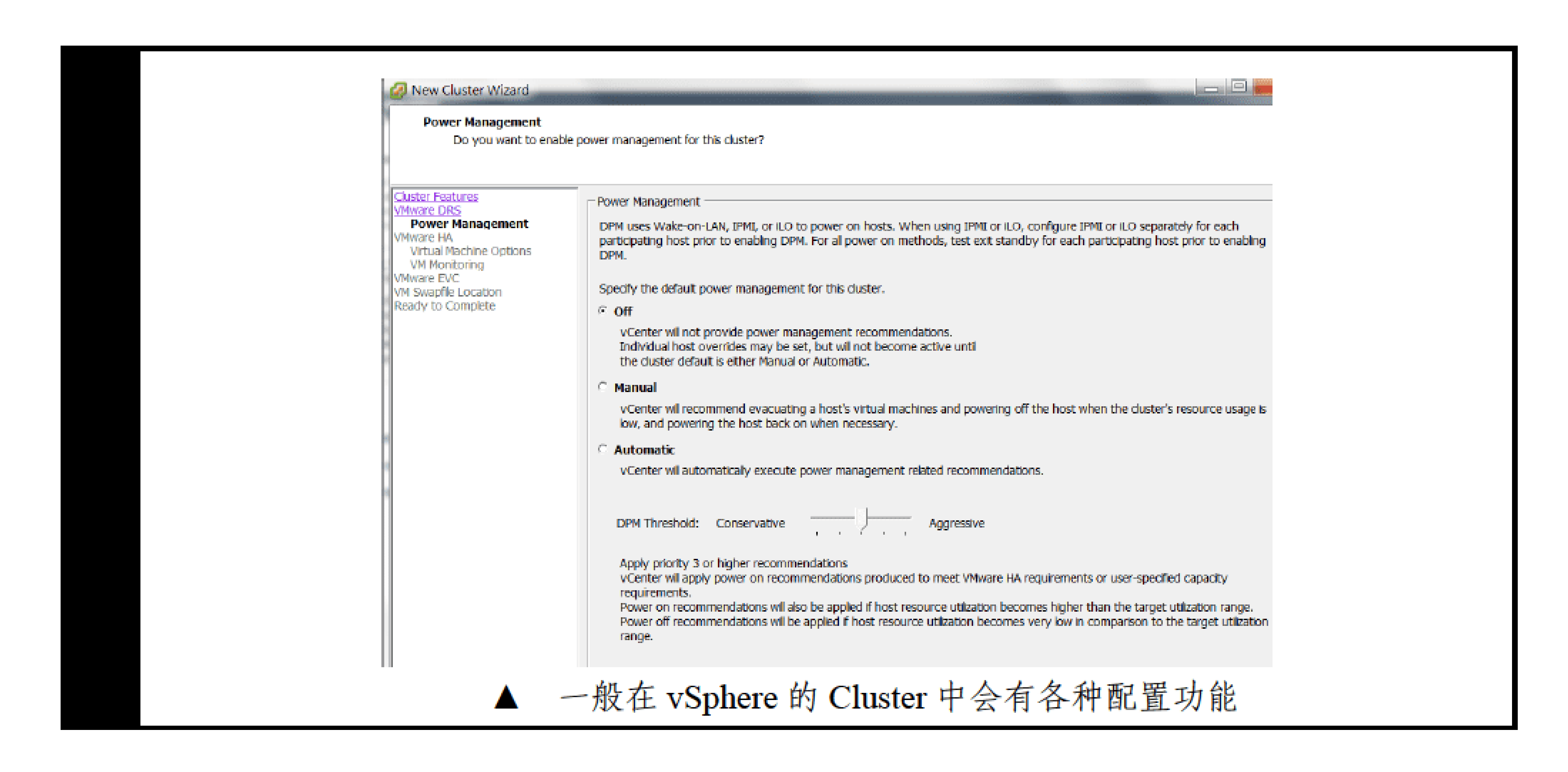
一提到 vSphere 的强大功能，大部分的人立马就想到 VMotion。VMotion 目的十分简单，就是当有一台物理的 ESX 服务器停机，其上的虚拟机可以在最快的时间内移动到另一台没有停机的 ESX 物理服务器上。事实上，在 Windows Server 2008 上的虚拟机转移，只是利用 Windows Server 2008 本身的 Failover Cluster 功能来转移虚拟机，当时被称为 Quick Migration。但由于微软竞争对手 VMware 的 VMotion 实在是太强悍了，因此 Live Migration 才在 Windows Server 2008 R2 上弹出。



▲ 在 Windows 2008 的 Hyper-V 中就有动态转移，但配置也很复杂

VMotion 在上一代的 ESX3.5 服务器中就已经弹出，这一代的 vSphere 4.0 的 VMotion 功能上更强，也支持了更直觉性的操作。我们就在这一小节中，先理解 VMotion 的原理，并且以上一章创建的示例环境来具体操作 VMotion。

> **注意**
> **VMotion 是一切 vSphere 高级功能的基础**
> VMware 中的 vSphere 企业解决专案提供的高级功能如 DRS、HA 等，事实上都是以 VMotion 为基础，再加上 vSphere 的集群功能组合而成的。因此在设计一个 vSphere 环境时，我们第一个实践的服务一定是 VMotion。当 VMotion 落实，再加上良好的集群设计，才有可能实现 DRS、HA 或是 VCB 等服务。



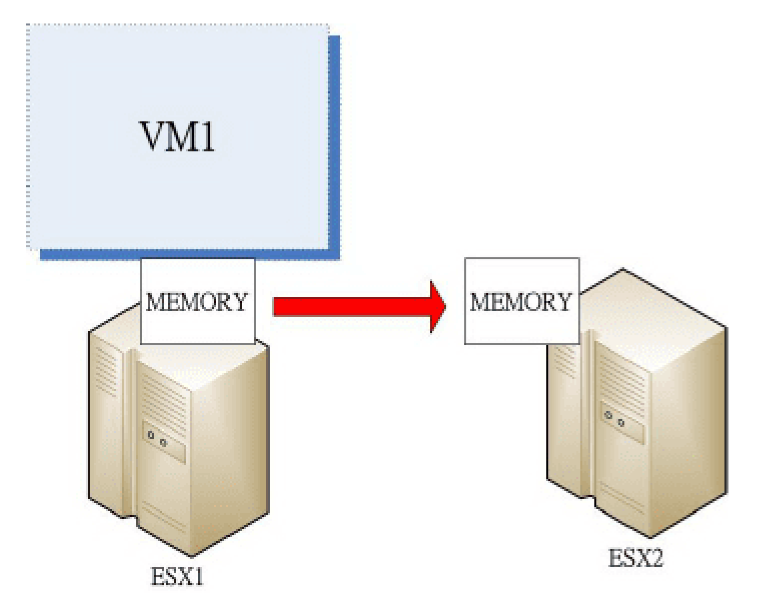
▲ 一般在 vSphere 的 Cluster 中会有各种配置功能

15.2.1 VMotion 的原理

事实上，VMotion 和 Hyper-V 的 Live Migration 可说是使用完全一样的原理。VMotion 在激活之后，系统也是先将在 192.168.1.81 上的虚拟机的内存状态克隆到 ESX2 上，再从独立的共享存储接管虚拟机硬盘文件（VMDK），当所有的动作落实后，就在 ESX2 上激活这台虚拟机。但 VMotion 的动作和速度，远非 Live Migration 可比拟，而 VMotion 也必须有相当的准备动作才能进行，我们就来看看 VMotion 的进行步骤和原理。

1. 内存的克隆

当具有权限的用户在 vCenter Server 中激活了由 192.168.1.81 转移到 ESX2 上的 VMotion 时，此时虚拟机还在常规使用的情况。来源的物理服务器 192.168.1.81 立即将本身使用中虚拟机的内存分页克隆到目的物理服务器 ESX2 上。在克隆的过程中，192.168.1.81 上的虚拟机仍然继续保持服务状态（如提供客户端的终端服务）。事实上在克隆的过程中，192.168.1.81 上虚拟机的内存分页还是有可能会改变。ESX 服务器会将这些改变表项下来，这些改变的日志称为内存对应图（Memory Bitmap）。内存对应图并不会表项内存中的内容，反而是表项内容发生改变的内存的位置，称为“脏内存”（Dirty memory）。

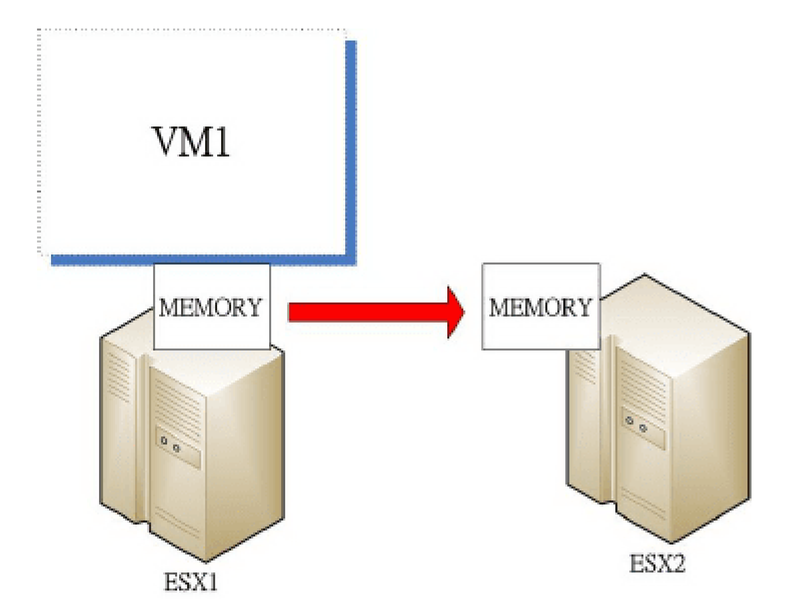


▲ 先克隆内存的位置

2. 虚拟机进入等待状态

当整个虚拟机的内存内容被移转到目的 ESX 服务器 ESX2 之后，在 192.168.1.81 上的虚拟机就会进入静止状态（quiesced）。进入静止状态的虚拟机无法再提供服务，如终端服务，但这台

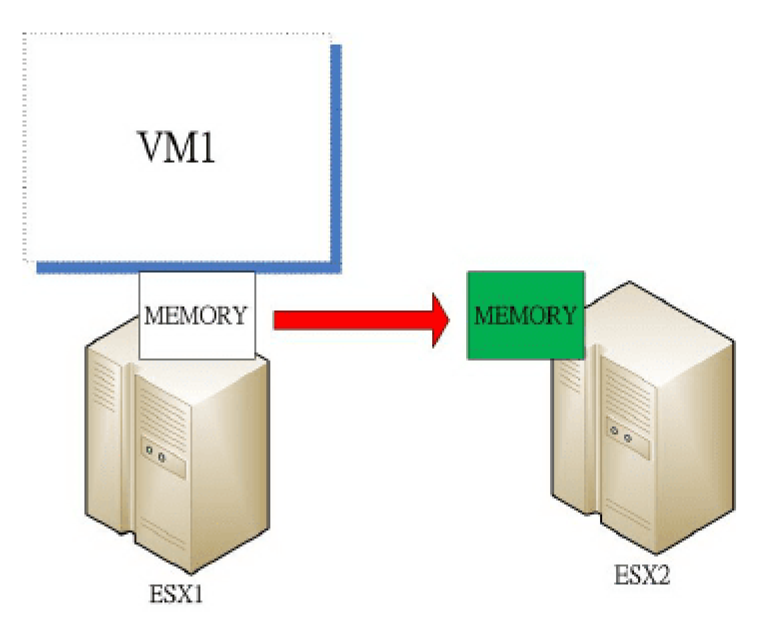
虚拟机的数据仍然保持在内存中。此时系统会将内存对应图从 192.168.1.81 克隆到 ESX2 上。



▲ 原始主机上的虚拟机会进入等待状态

3. ESX2 可读具体内存内容

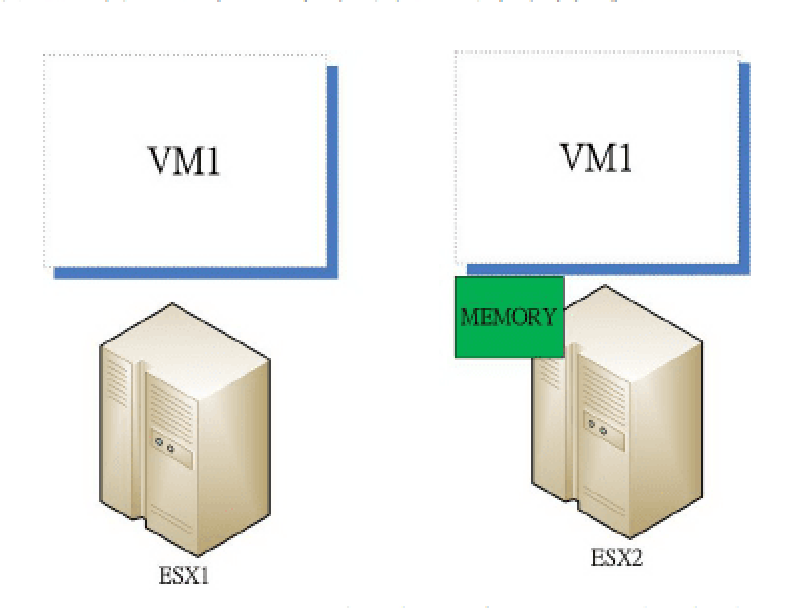
当 ESX2 获取内存对应图之后，就会依照图中的内存位置开始克隆每一个内存位置中的内容。这个动作就好像 192.168.1.81 先将一个移交清单交给 ESX2，之后 ESX2 就会照着移交清单中的条例，真正将给定的数据从 192.168.1.81 搬到对应的 ESX2 的内存中。



▲ 将内容移交清单给 ESX2

4. 激活 ESX2 上的虚拟机

当 192.168.1.81 上虚拟机进入静止模式之前的所有内存状态都被克隆到 ESX2 上之后，ESX 上就会根据内存对应图的状态激活这个虚拟机（因为 VMDK 都在共享存储中）。由于数据早已经存在内存中，因此 ESX2 上的虚拟机并不是“重新激活”，而是“激活后直接对应到内存上”。这个观念有点像较新计算机的“状态压入硬盘”功能，只是这里是两台物理服务器之间的切换，ESX2 上的虚拟机不是重启，而是对应到一个固化的内存状态。

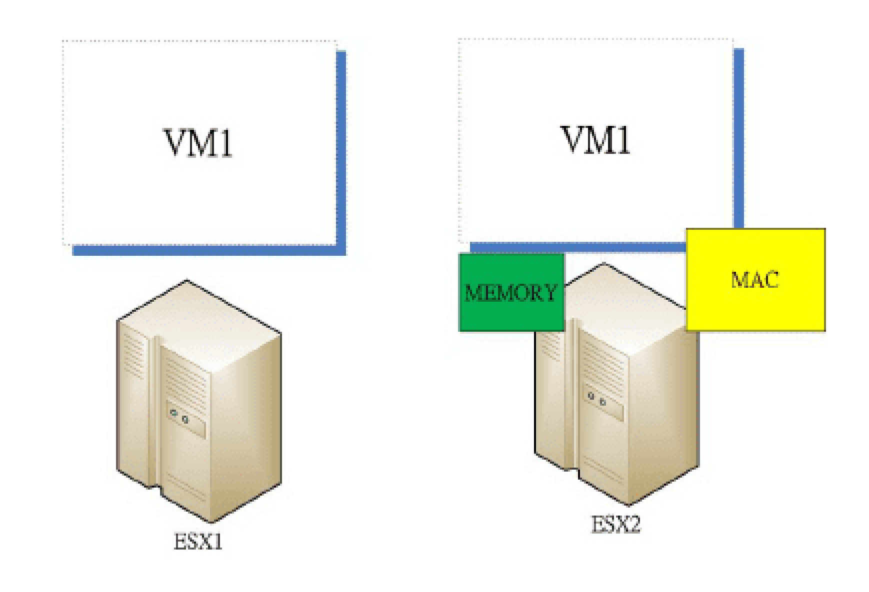


▲ 激活 ESX 上的原始虚拟机，可读其内存位置

5. 网卡 MAC 的重新对应

此时系统会激活一个称为“反向位置解析通信协议”（Reverse Address Resolution Protocol，RARP）的动作，在物理的交换机上，将原来 192.168.1.81 所代表的 MAC 位置换成 ESX2 的 MAC

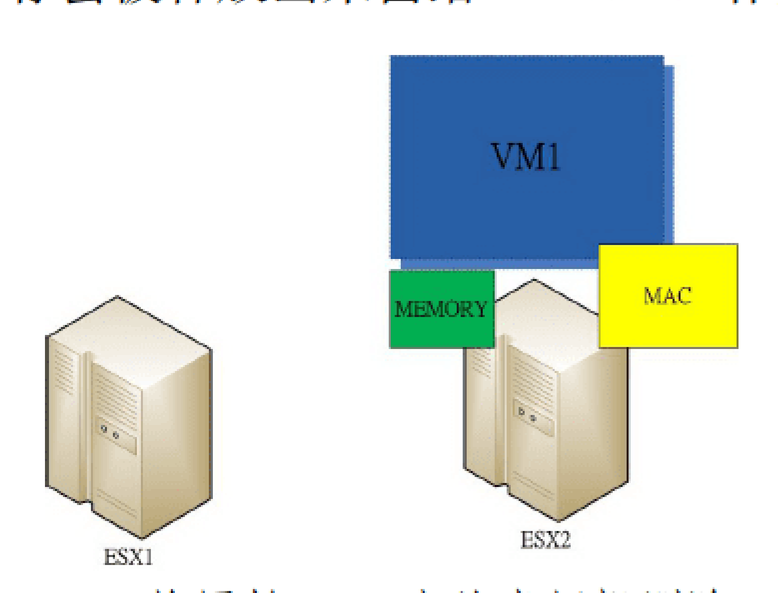
位置。这么做的原因是要让网卡原来访问 192.168.1.81 上虚拟机的客户端，在不知情的情况下，将交换机上的报文重新定位到 ESX2 上的虚拟机。



▲ 对应 MAC 地址

6. 删除来源机上的内存内容

当数据克隆完毕，网络 MAC 地址也指向成功之后，虚拟机使用的内存就会在来源的 192.168.1.81 上被删除，并且内存会被释放出来留给 VMkernel 作为日后的使用。



▲ 将原始 ESX 上的虚拟机删除

15.2.2 VMotion 的基本条件

除了安装好 vCenter Server 以及对应的环境之外，我们要在 vCenter Server 中实行 VMotion 必须满足三个前提条件。分别是 ESX 服务器部分、网络交换机部分以及虚拟机部分。

1. ESX 服务器部分

ESX 服务器群的配置是 VMotion 成功的关键，因此在运行 VMotion 之前，我们必须针对两台服务器进行遍历。由于我们在上一章已经落实了所有的环境配置，因此我们只要针对物理服务器的部分做遍历。

▶ 使用 VMotion 的 ESX 服务器检查项目

- 参与 VMotion 的所有 ESX 服务器要联上同一个共享的外部存储。本例中两台 ESX 服务器都连上了 QNAP 的 iSCSI 或是 NFS，而实验环境也是连上同一个 QNAP 的 NAS。
- 多台参与 VMotion 的 ESX 服务器必须使用完全一样的硬设备，尤其是 CPU。
- 同样的品牌（都是 INTEL 或是 AMD）。
- 同样的型号（都是 Xeon 5530）。
- 同样的内核数（都是四核）。
- 同样的命令集（都支持 SSE4、支持 NX）。
- 如果是 64 比特的虚拟机，都要支持硬件的虚拟机命令（INTEL-VT 或 AMD-V）。

为了不要被排错误占用太多宝贵的时间和工夫，在准备 VMotion 的运行环境时，一定要让

所有参与 VMotion 的机器都使用完全一样的厂牌、型号、CPU、内存大小、HBA 等，所有的机器都要完全一样（Completely identical）才行。

2. 网络的需求

要落实 VMotion 的动作，网络上的配置比 CPU 更为重要。两台参与 VMotion 的主机，一定要拥有完全一样名称的 vSwitch，一样名称的 VMkernel Port，这些都是 VMotion 的基础，必须理解才能进行 VMotion。我们在此就来遍历一下运行 VMotion 时，两台 ESX 上的网络配置。

▶ 运行 VMotion 的两台计算机网络配置

（1）两台计算机必须连上同一个物理网段，本例都是 172.17.23.X/255.255.254.0 网段。

（2）在作为 VMotion 的网卡 vnmic1 上，要配置一个完全一样名称的 vSwitch，如 vSwitch0。

（3）在 vSwitch0 之上，都要有一个 VMkernel 通信端口，用来作为 VMotion 之用。其名称要完全一样，如全大写的 VMOTION。注意大小写要完全一样。

（4）在给虚拟机用的通信端口组上，其名称要完全一样，如 VM。这样在虚拟机从 192.168.1.81 移转到 ESX2 时，其网卡也能常规使用。

（5）在 192.168.1.81 上所有 vSwitch、通信端口组的数量、功能和名称，ESX2 上要完全拥有一模一样的。如果 192.168.1.81 有连上某个 VLAN，ESX2 也必须连上一样的 VLAN。

（6）确定两台 ESX 能找到对方。你可以进入任何一台 ESX，并且在主控台中 ping 对方的位置。ESX 提供了一个 vmkping 工具，主要用来 ping VMkernel 这个通信端口，只要能 ping 通对方，两台 ESX 在 VMotion 中就没有什么疑义了。

```
VMware ESX Server 3
Kernel 2.4.21-57.ELvmnix on an i686

esx2 login: root
Password:
Last login: Fri Mar 20 22:46:11 on tty1
[root@esx2 root]# vmkping 10.0.0.10
PING 10.0.0.10 (10.0.0.10): 56 data bytes
64 bytes from 10.0.0.10: icmp_seq=0 ttl=64 time=3.847 ms
64 bytes from 10.0.0.10: icmp_seq=1 ttl=64 time=1.546 ms
64 bytes from 10.0.0.10: icmp_seq=2 ttl=64 time=1.700 ms

--- 10.0.0.10 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 1.546/2.364/3.847 ms
[root@esx2 root]# _
```

▲ 必须使用 vmkping 通对方才行

3. 虚拟机的需求

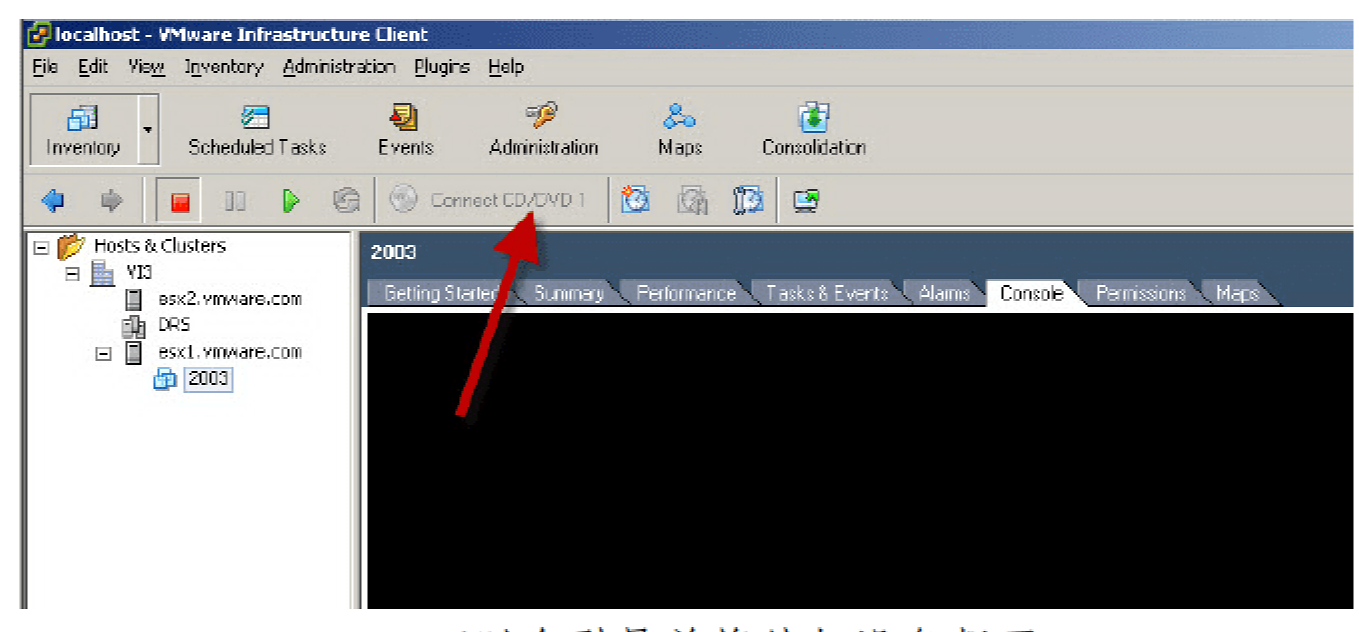
要运行 VMotion 的虚拟机，也必须遵守 vSphere 的规定。最重要的就是不能让这个虚拟机在两台物理服务器之间起到差异，下面就是详细的说明。

▶ 运行 VMotion 虚拟机的需求

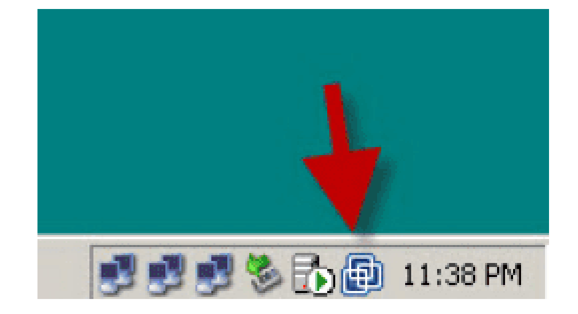
- 虚拟机不可以连上任何只有某台 ESX 主机才辨识出来的设备。如光盘机、软驱或是 USB、串口等。如果要做 VMotion 的虚拟机必须拥有这些设备，那么在运行 VMotion 之前，将这些设备的 Connected 选择撤销。
- 虚拟机不可以连入没有接上物理网络的虚拟交换机。
- 虚拟机不可将 CPU 的权限配置到某一个物理 CPU 上。
- 虚拟机不可以对应到微软集群服务的 Raw 格式磁盘机上。
- 和虚拟机有关的所有文件，包括 VMX、LOG、随机内存文件（NVRAM）、VMDK、LCK

等都必须放在共享的存储设备上。

- 一定要安装 VMware Tools。



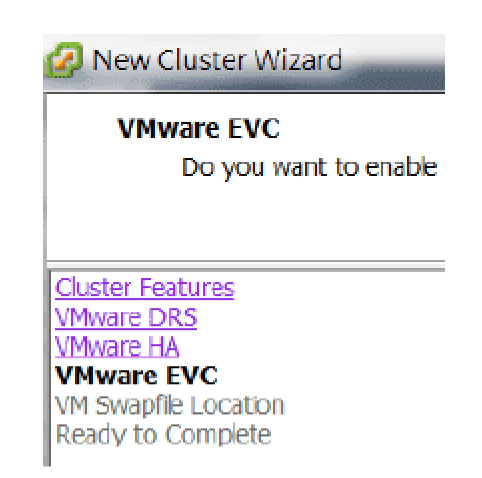
▲ 可以在引导前将外加设备断开



▲ 一定要安装 VMware Tools

15.2.3 理解什么是 EVC 模式

当我们在一个集群中创建多台 ESX 主机时，意味着将会使用到较高级的功能，如 HA、DRS 等，那么你一定需要 VMotion 来落实这些功能。但在多台 ESX 中，如果 CPU 的型号差距太大，VMotion 很有可能要花上很多时间落实，甚至是无法实践，因而引起更多高级功能失灵，因此为了确保在一个 Cluster 中的所有 ESX 主机能顺利激活 VMotion，VMware 在 vSphere 中使用一个 EVC 模式来进行 VMotion 的规范化。



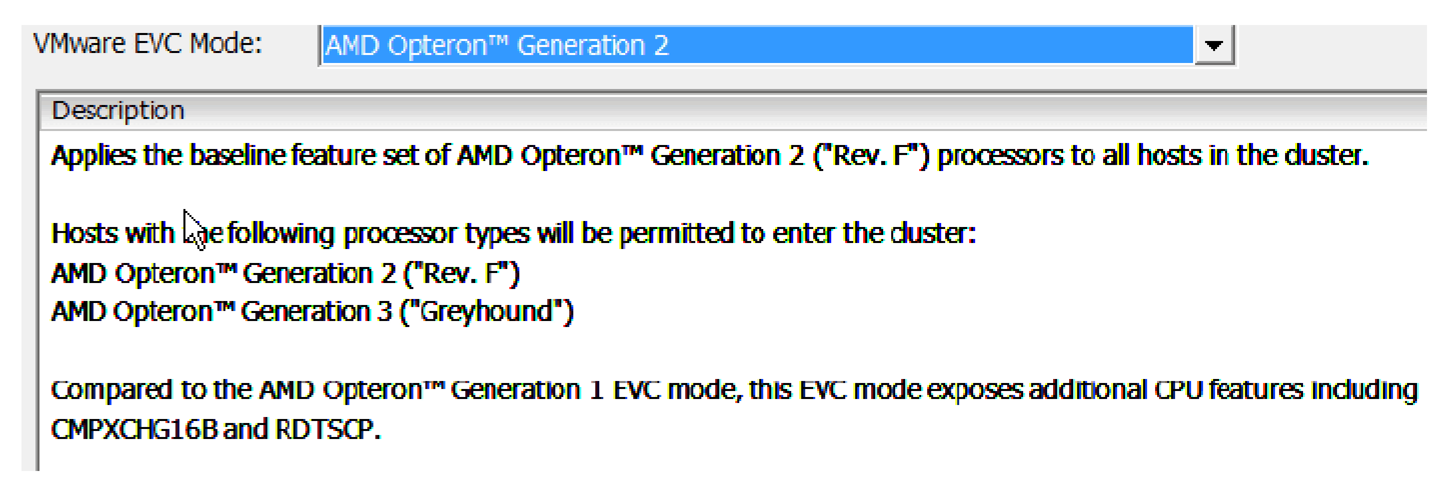
▲ 在配置 Cluster 时，可以选择 EVC 模式

1. 什么是 EVC 模式

EVC 模式是 Enhanced VMotion Compatibility 的缩写，当在一个 Cluster 中的 EVC 模式被激活后，在加入 ESX 主机时，如果这台主机不匹配当前 EVC 模式的配置，就无法将这个主机加入。

2. EVC 模式的用途

EVC 模式的作用是让 vSphere 来帮你判断新加入的主机是否能胜任 VMotion 的能力。由于当前 CPU 的产品种类太多，因此需要事先判断是否能使用 VMotion，如果要让用户自身查证将会是很大的麻烦。EVC 让 vSphere 来做这些判断，不但可以确定兼容性，还可以让整个 Cluster 在进行高级功能时，能保障速度与兼容性不会受影响。



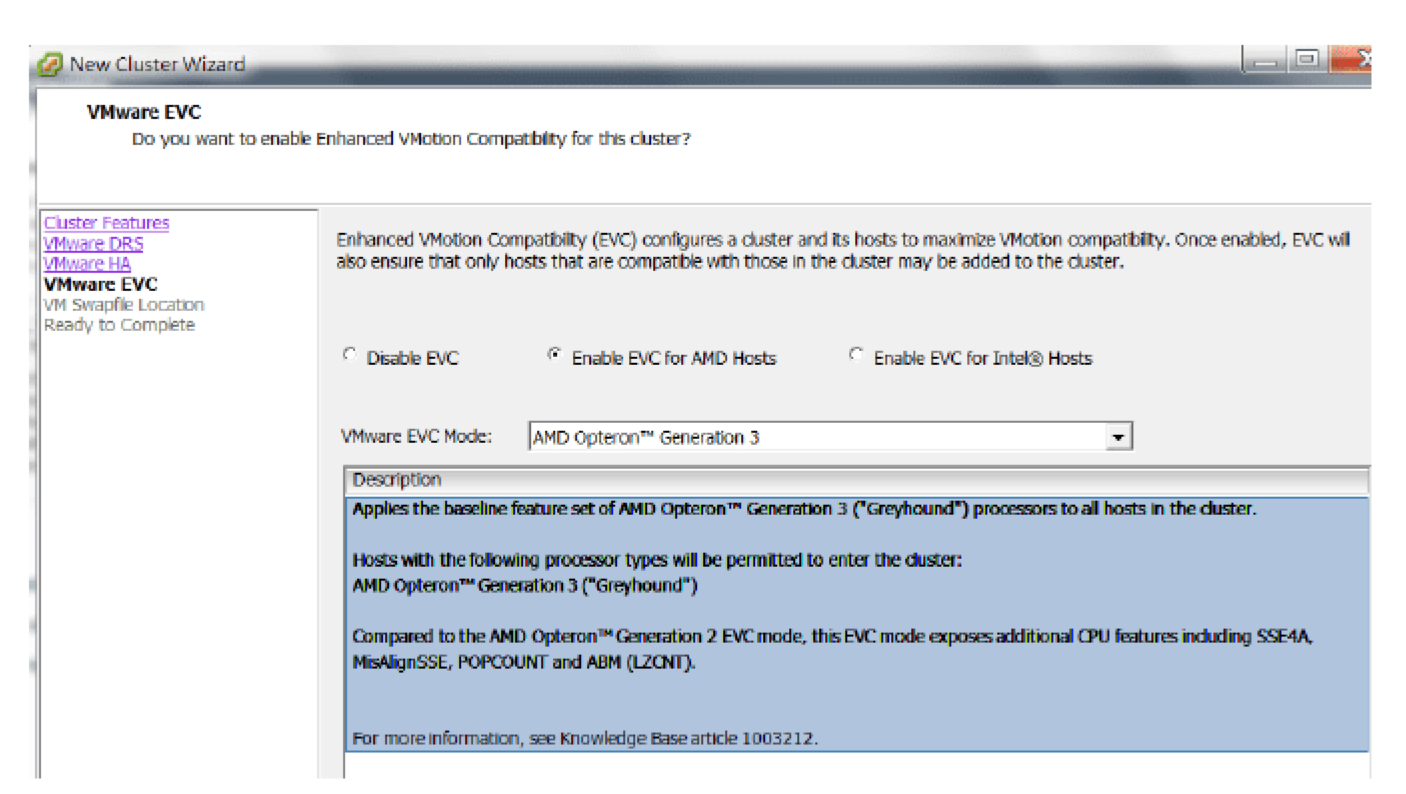
▲ 如果加入 vSphere 的主机有各种样的品牌，使用 EVC 可以确保兼容性

3. EVC 模式的种类

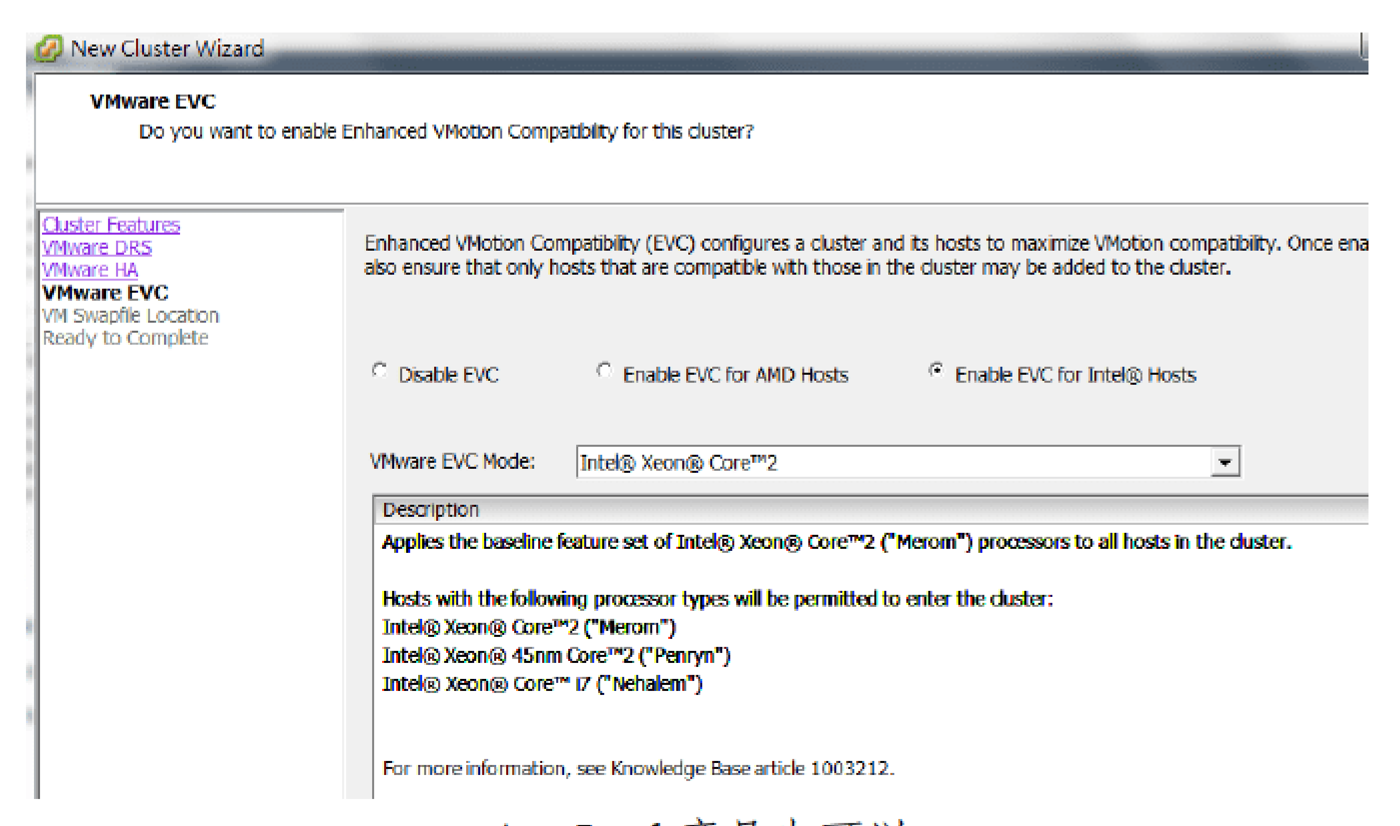
在 vSphere 下，EVC 模式一共有三种，下面就分别来看看。

▶ vSphere 三种 EVC 模式

- Disable：不遍历兼容性，这是兼容性最高的 VMotion 模式，可以确保不同的 CPU 产品也能进行 VMotion。但速度和功能就无法确定一定能达到要求了。
- Enable EVC for AMD Hosts：这是 AMD CPU 的选项，当选择这个选项之后，只有 AMD Opteron 的 CPU 才能加入 Cluster。由于 Opteron 有 3 个版本（Generation 1, 2, 3），因此当你选择第一项时，是所有的 3 个版本都可加入，选择第二项是只有 2、3 可以加入，选择第三项只能加上第 3 版，会将范围越缩越小。
- Enable EVC for Intel Hosts：这是 Intel CPU 的选项，选择这个选项之后，只有 Intel 的特殊 CPU 才能加入 Cluster。这里和 AMD 一样，也是越小范围越严格，选择第一项，Intel 的 Core 2、45nm Core2 及 i7 都可以加入，选择第二项只能加入 45nm Core2 和 i7，选择第三项，就只能加入 i7 了。



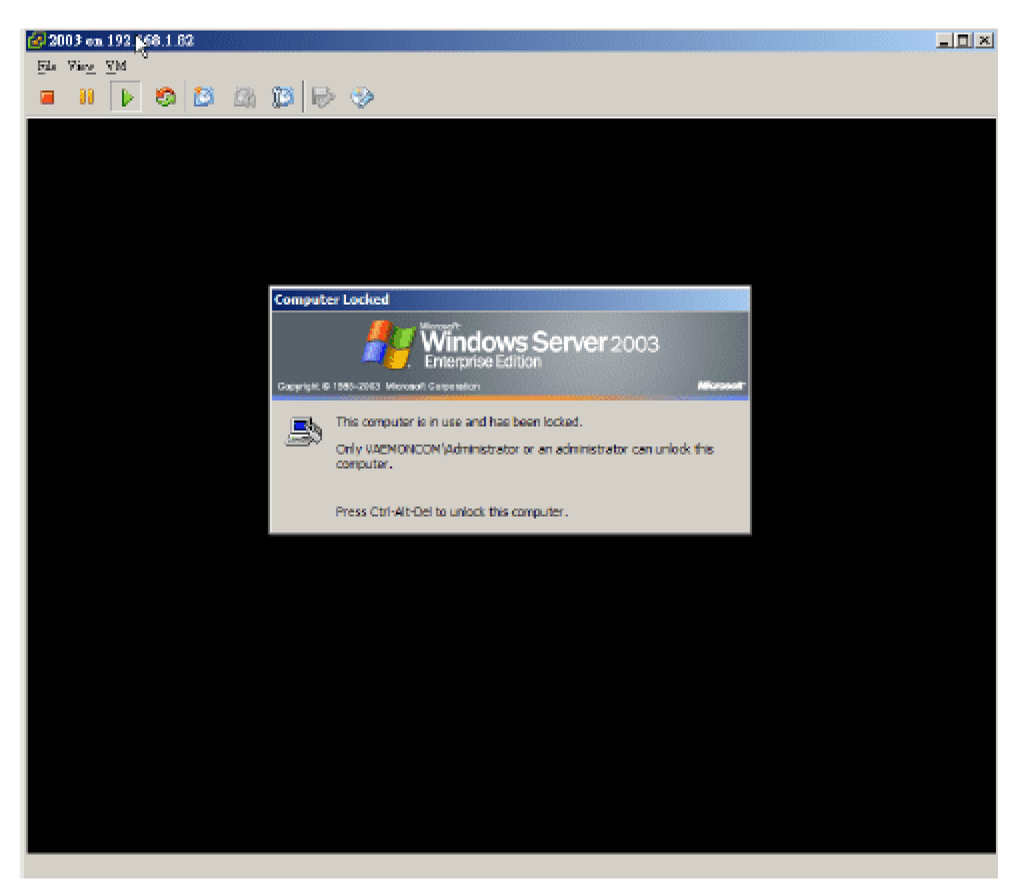
▲ 在 AMD 的兼容性上也可以选择更细一步的产品



▲ Intel 产品也可以

15.2.4 立即实践 VMotion

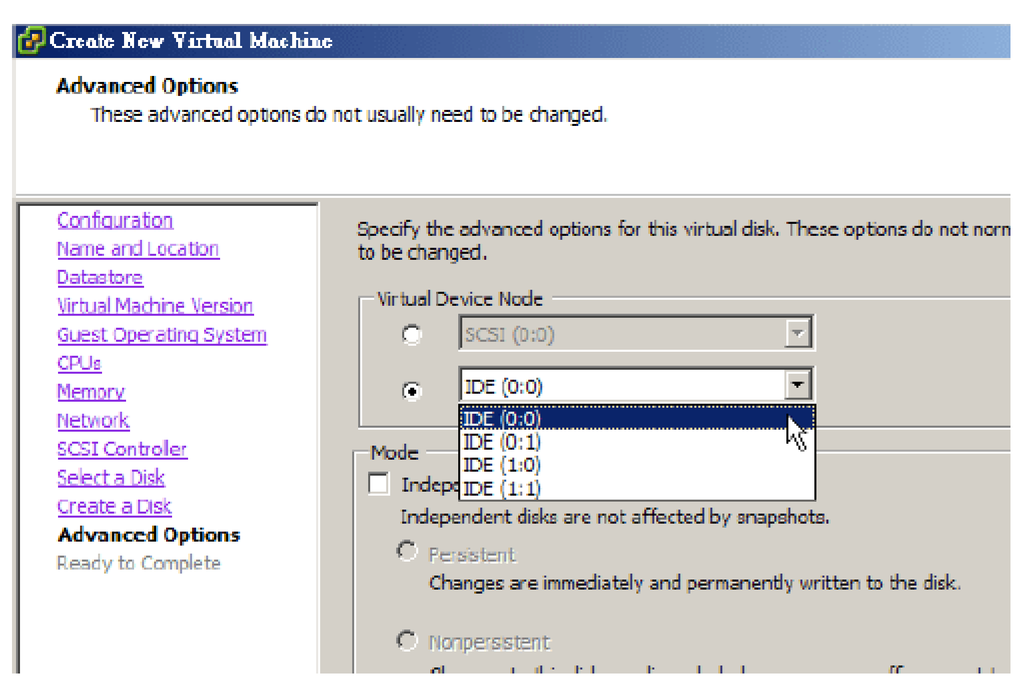
在落实了所有 VMotion 的配置之后，读者们一定认为 VMotion 十分复杂吧！事实上，当你的配置完全正确时，激活 VMotion 竟然只是几个鼠标动作就可以落实的事！当然在使用 VMotion 之前，我们得安装一个操作系统才能进行 VMotion。



▲ 安装一个 VM，运行 2003

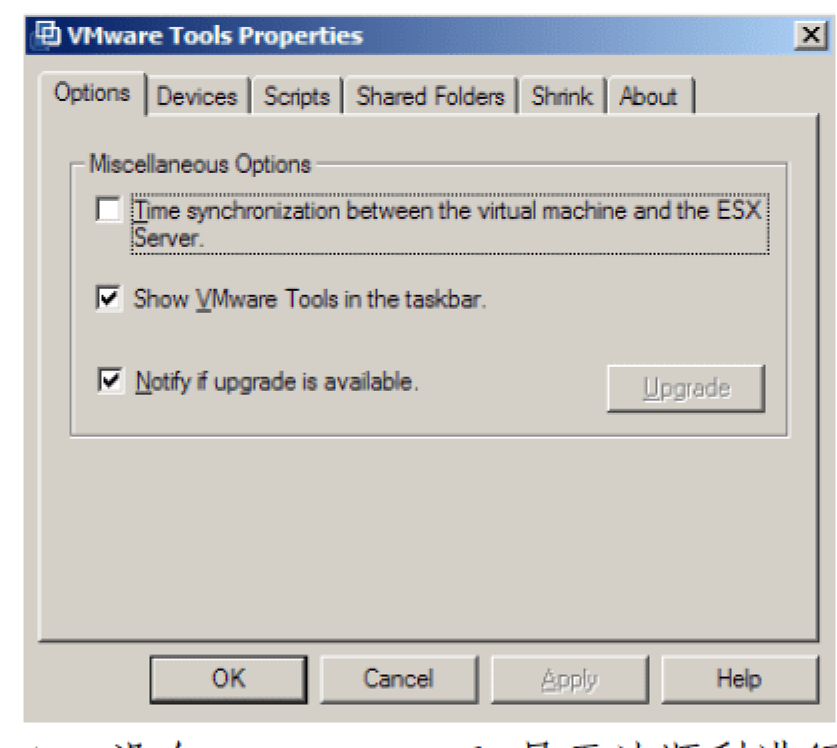
1. 装虚拟机操作系统

在 ESX 中安装 Windows 操作系统时，由于 ESX 本身提供的硬盘为 SCSI，因此安装 Windows XP 时必须使用 ESX 中自身附带的 SCSI 驱动，并且给定到软盘中。而 Windows Server 2003 直接支持 SCSI，因此我们就在 ESX 中安装一个 Windows 2003 较为方便。

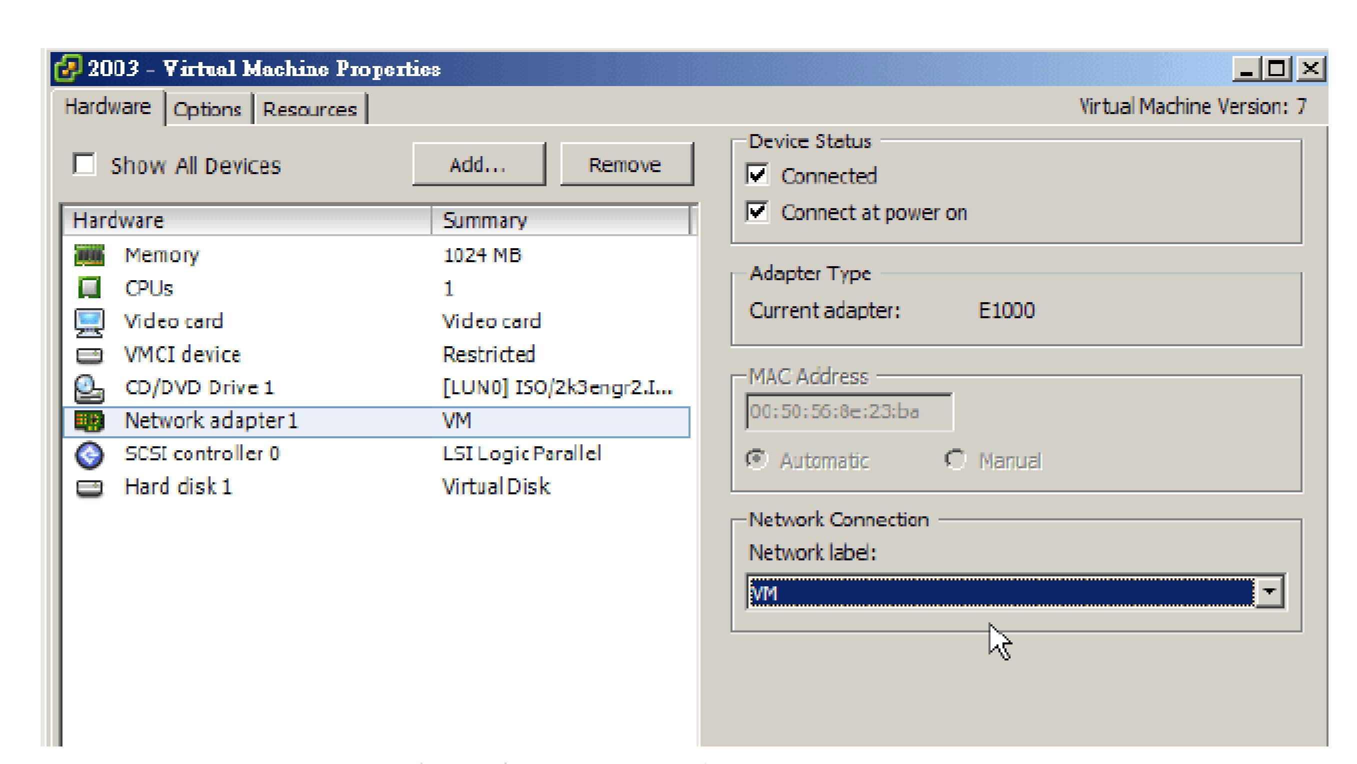


▲ 旧版的 vSphere 不支持 XP 的 IDE 模式，现在已经可以了

而安装 ESX 中虚拟机的步骤相当简单，读者们一定已经驾轻就熟了，只要在安装 Windows Server 2003 时，注意选择网卡，一定要选择刚才配置在 vmnic0/vSwitch0 上的 VM 这个通信端口组。其他就照常安装即可。在安装完毕后，别忘了安装 VMware Tools。



▲ 没有 VMware Tools 是无法顺利进行 VMotion 的



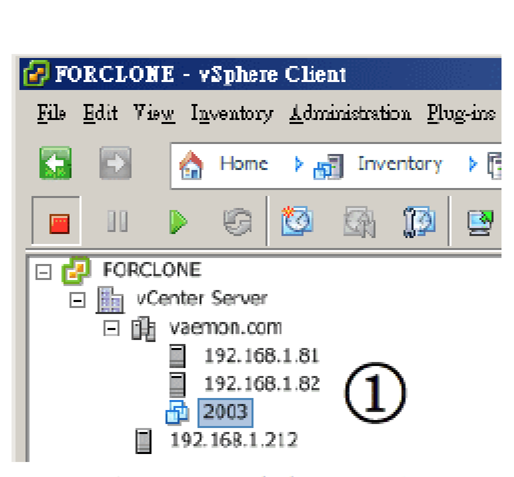
▲ 安装虚拟机时不要忘了选择在 vmnic0 中创建的虚拟机通信端口组

2. 先从简单的非实时转移开始

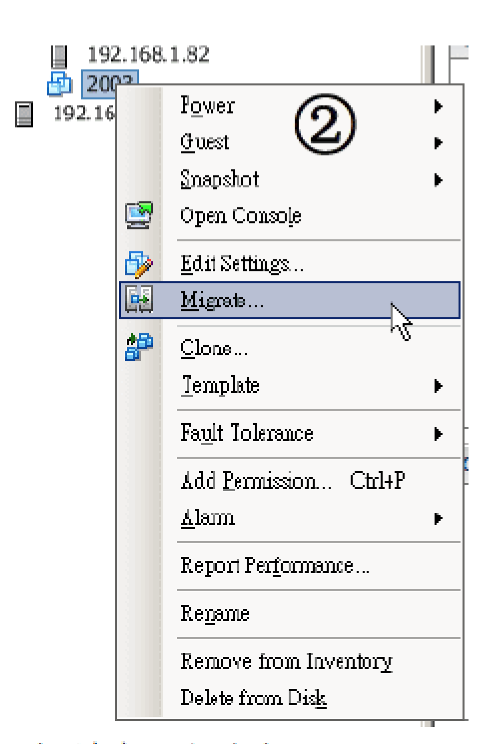
在进行 VMotion 之前，我们可以先来看看旧的做法是如何进行的。举例来说，如果有一台物理机 192.168.1.81 上运行了一个虚拟机 Windows Server 2003，名字叫 2003，其提供了终端服务，当我们要针对 192.168.1.81 进行维修时，必须先将 192.168.1.81 上的 2003 关机，移到常规使用的 192.168.1.82 上。在没有 VMotion 时的做法如下。

▶ **非动态转移的做法**

1. 确定 192.168.1.81 上的虚拟机 2003 是关机的状态。
2. 直接将 2003 这个虚拟机从 192.168.1.81 上拖到 192.168.1.82 上，或是在虚拟机 2003 上右击，并且选择 Migrate 选项。

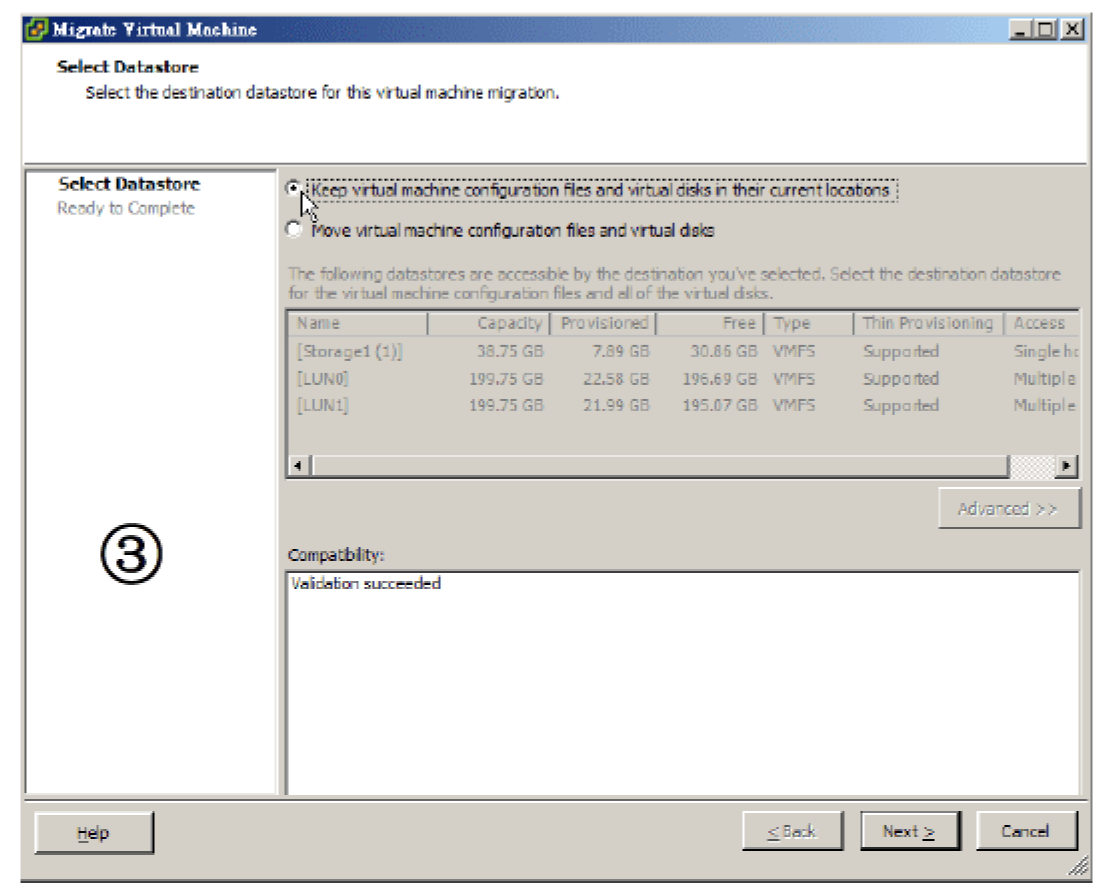


▲ 确定是关机的状态

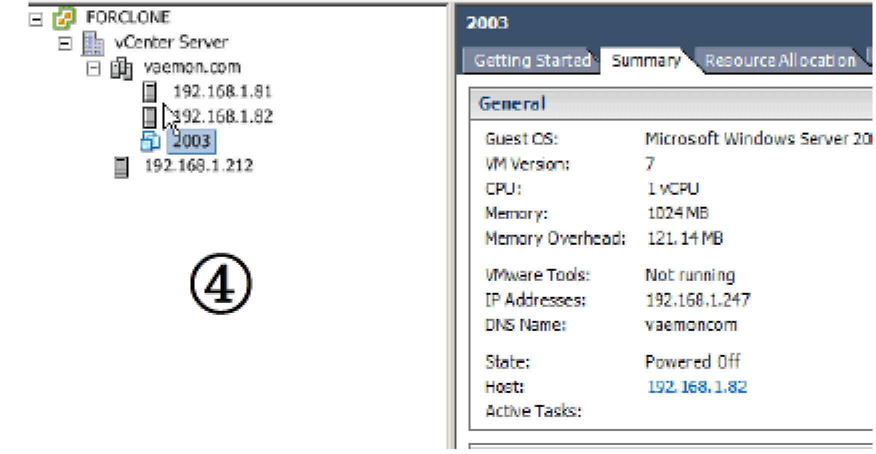


▲ 在关机时选择 Migrate

3. 选择要转移的机器。此时系统会询问是否要移动文件。由于我们所有的文件都放在独立的 iSCSI 上，因此没必要移动文件。
4. 由于没有需要克隆的内存，因此可以看到拖过去之后，虚拟机 2003 就弹出在 192.168.1.82 上了。



▲ 没有必要移动文件



▲ 弹出在 192.168.1.82 上了

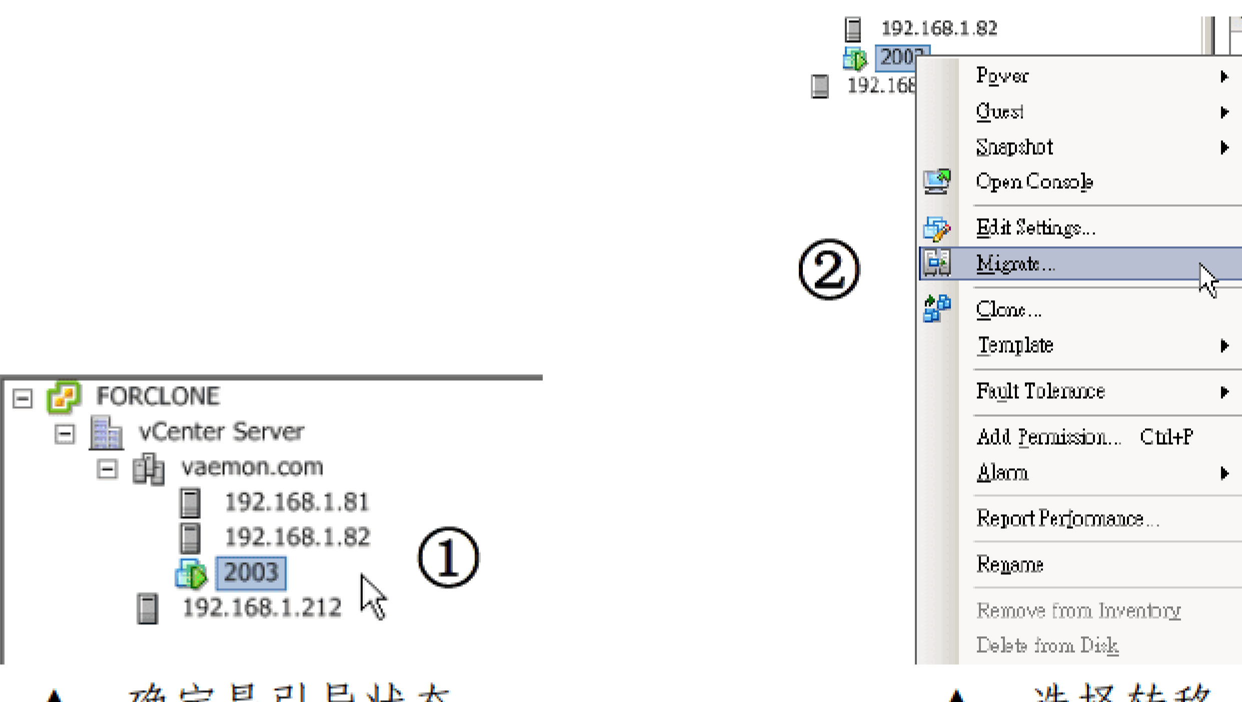
3. 激活 VMotion

终于到了等待已久的时刻了！VMotion 的激活热点，就是 192.168.1.81 上的虚拟机正在引导状态，在完全不关机，不影响用户使用的情况下，将虚拟机 2003 从物理服务器 192.168.1.81 移到物理服务器 192.168.1.82 上。我们就来看看 VMotion 是如何落实的。

▶ 具体操作 VMotion

1．确定虚拟机 2003 是在引导状态，并且 VMware Tools 在虚拟机已经正确安装并激活。

2．一样在虚拟机上按下鼠标右键，并且选择 Migrate 选项。

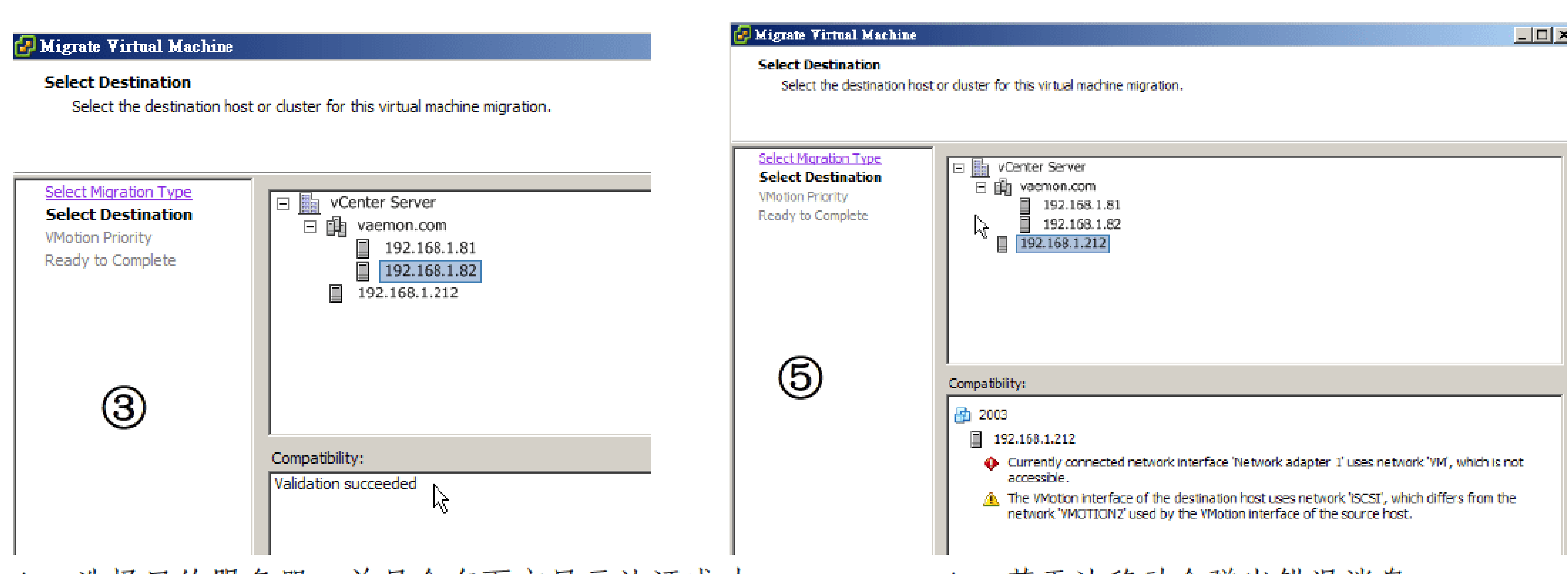


▲ 确定是引导状态　　▲ 选择转移

3．此时系统会要求选择转移方式，就像上面一样选择第一项 Change host。之后键入目的 ESX 服务器，我们就选择 192.168.1.82。

4．之后系统会确认握手是否成功，成功后单击下一步。

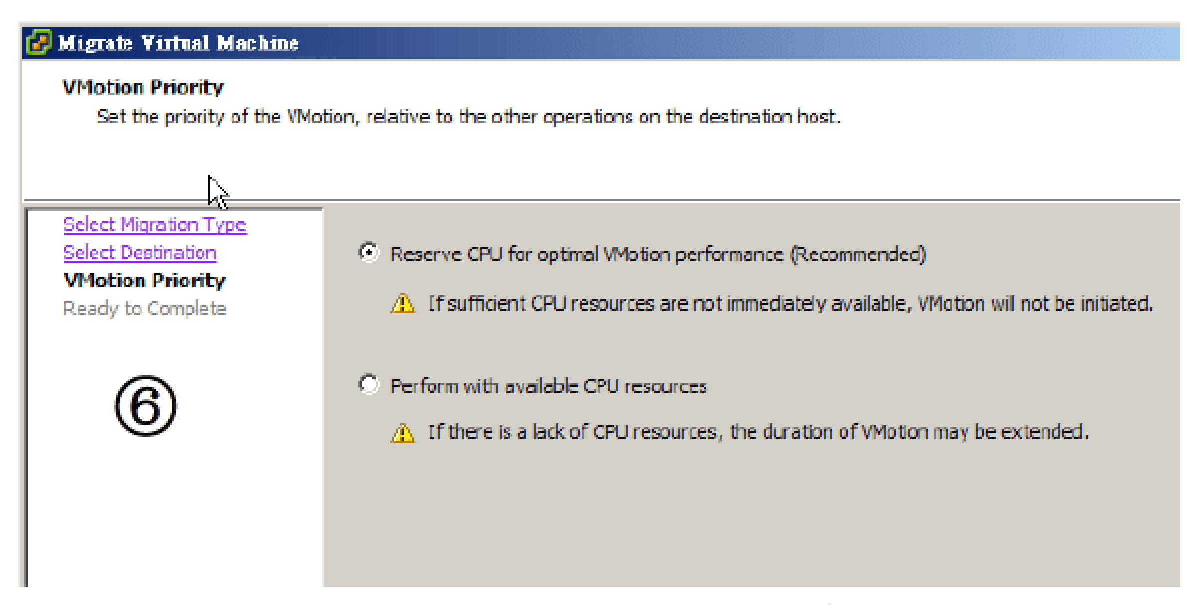
5．接下来系统会要求选择模式，我们选择第一项，之后单击 Next 按钮。

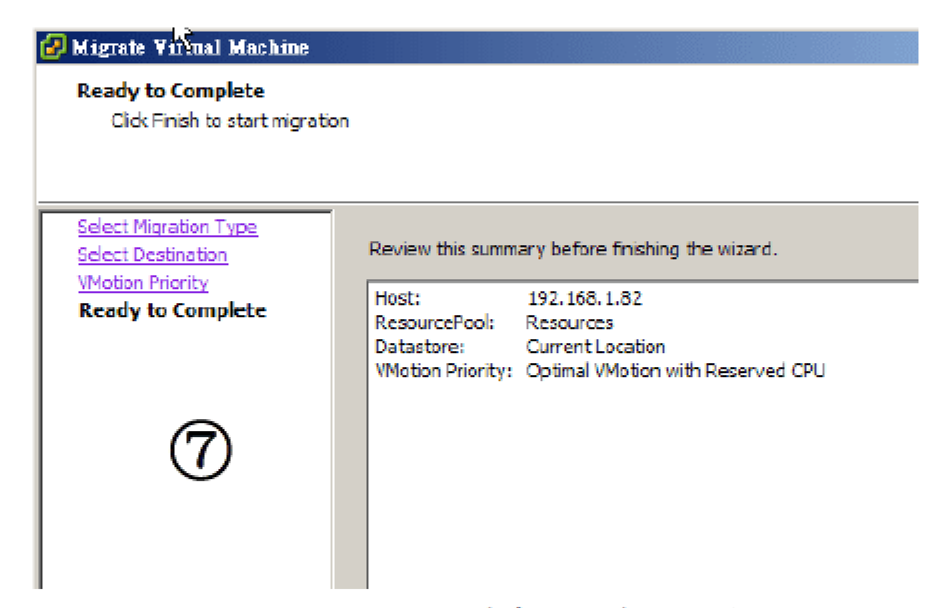


▲ 选择目的服务器，并且会在下方显示认证成功　　▲ 若无法移动会弹出错误消息

6．之后系统会要求配置优先序，使用高优先序可以让整个 VMotion 的速度加快，因此我们选择第一项。

7．之后系统就会再重新确定一次整个设置，单击 Finish 按钮之后就会开始转移。

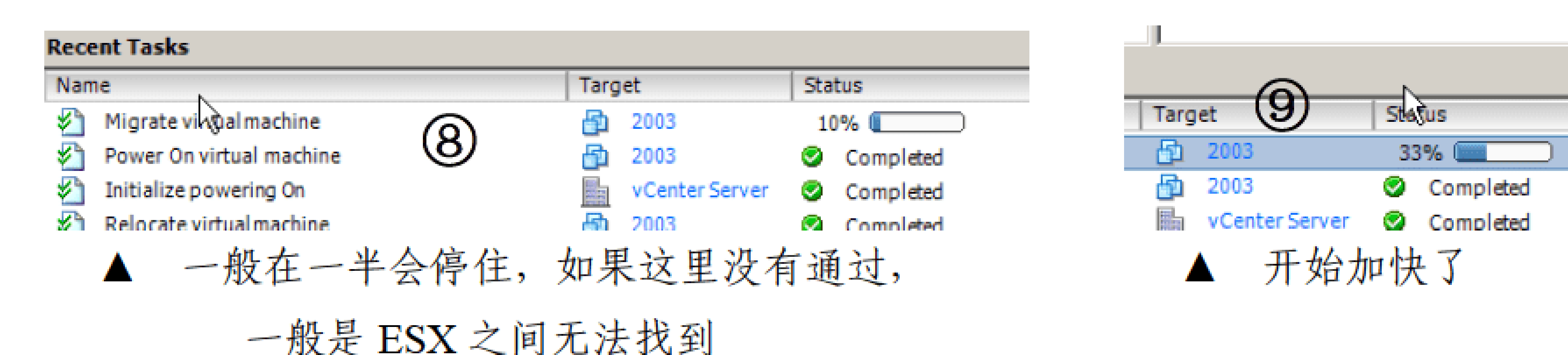




▲ 选择高优先序会让系统速度加快　　▲ 重新回车设置

8．此时下方的事件视图器会给出进度。一般来说，系统会在一半的地方停留较久，主要就是先进行内存的克隆。

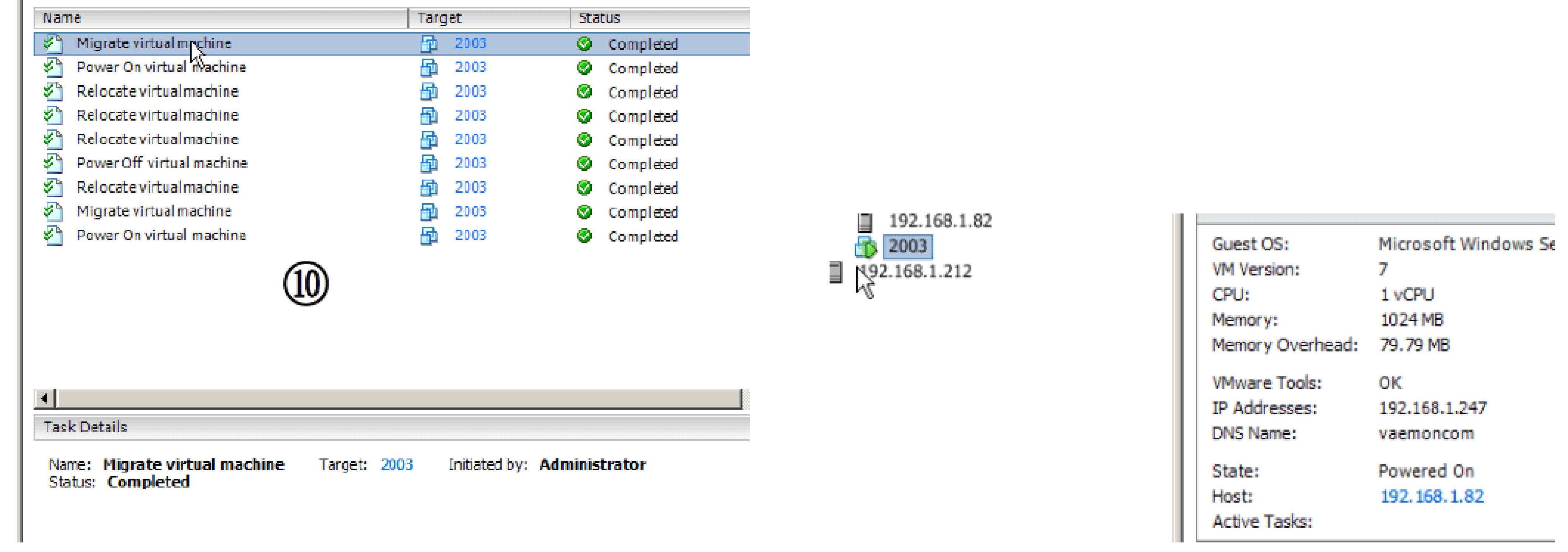
9．当突破 10%之后，系统的速度就会加快。



▲ 一般在一半会停住，如果这里没有通过，一般是 ESX 之间无法找到

▲ 开始加快了

10．当 VMotion 落实之后，事件视图器也会显示 VMotion 落实的画面。



▲ 事件视图器会说明 VMotion 落实

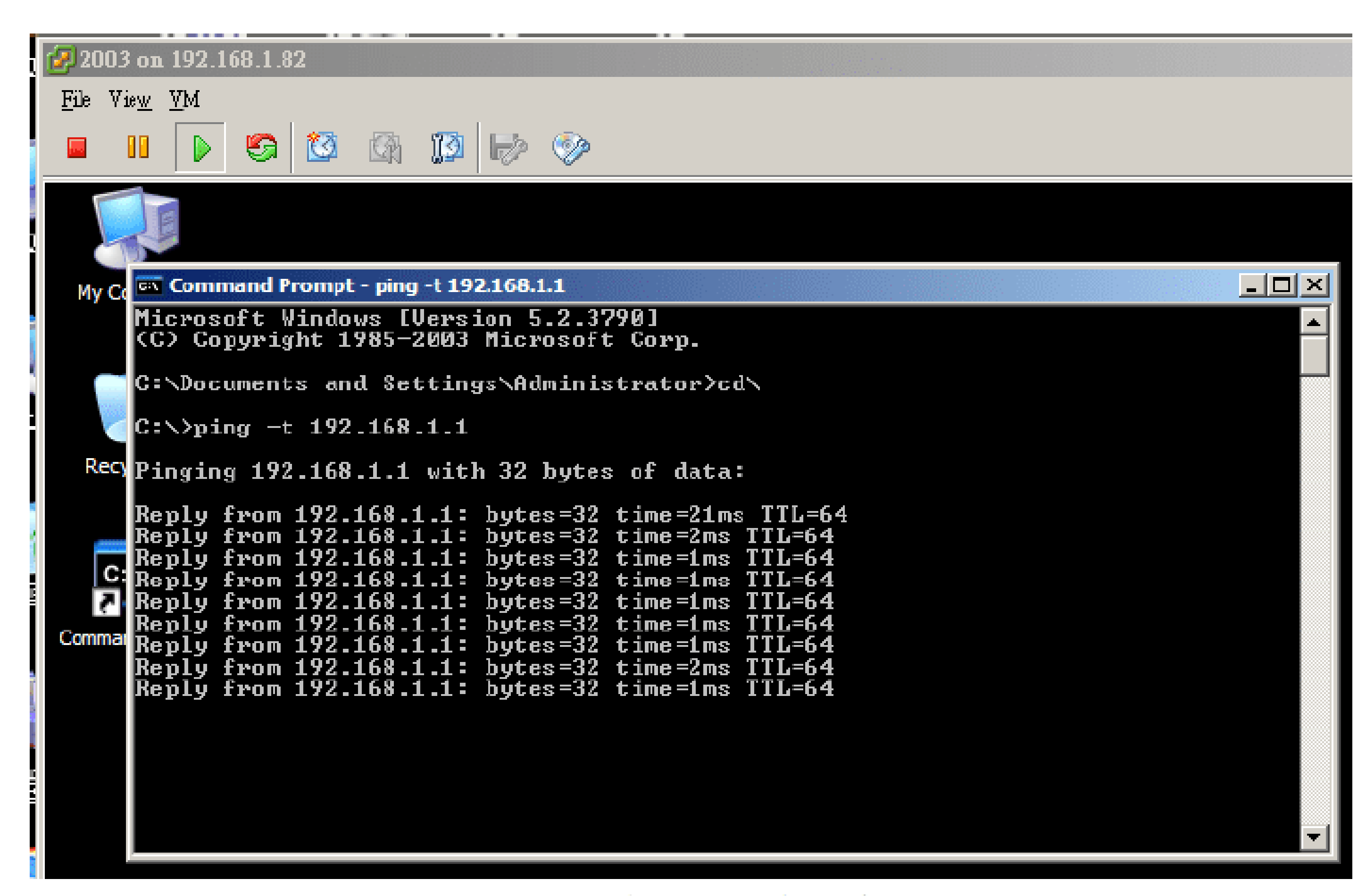
▲ 顺利移到另一台机器上了

4．近距离观察 VMotion

我们从 VC 主机来 ping 虚拟机 2003，并且直接观察 VMotion 的作用情况，看看在整个过程中，虚拟机 2003 是否有停机或是无法使用。

▶ 观察 VMotion 进行时虚拟机的状况

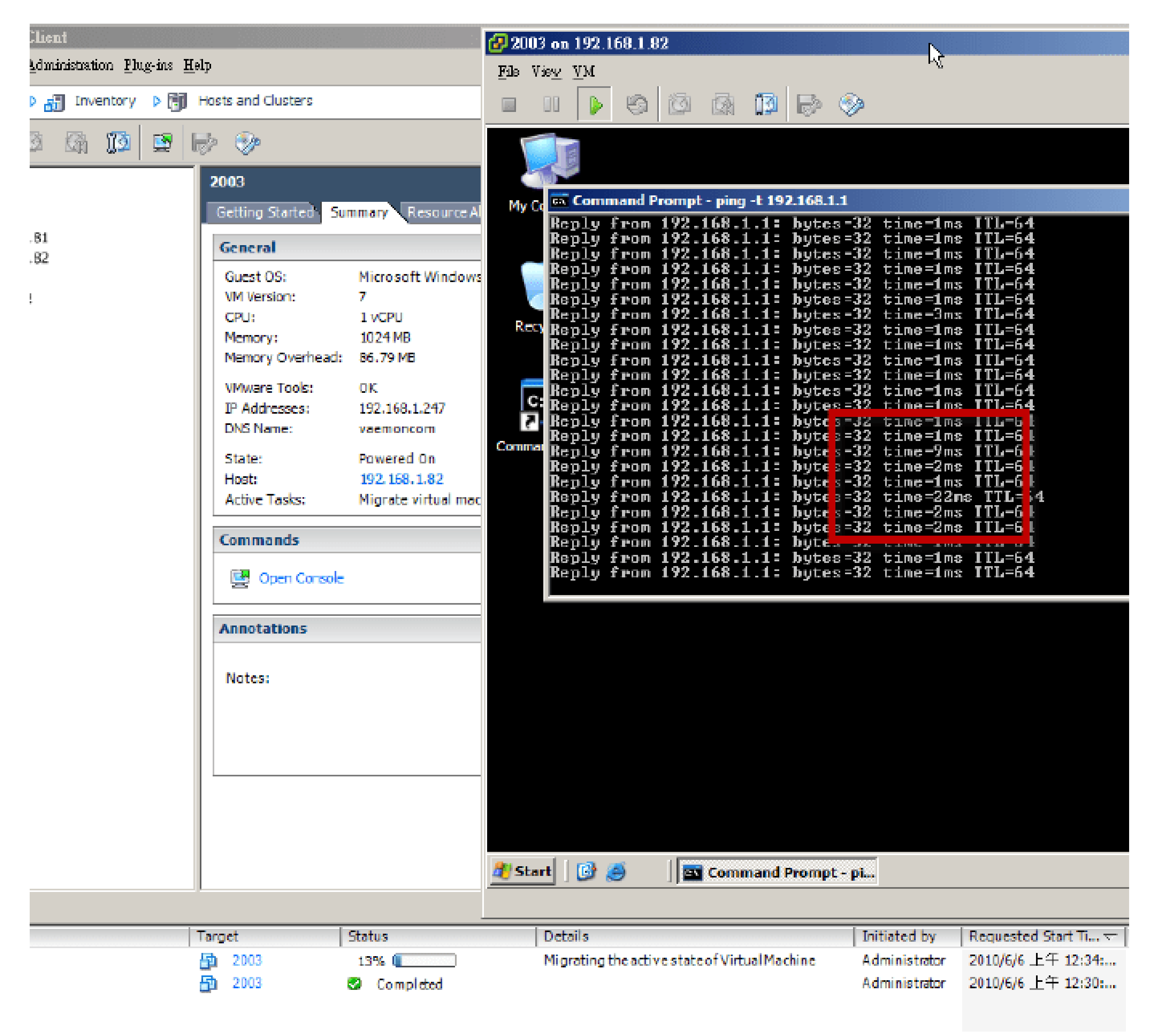
1．首先我们把虚拟机的 IP 设成和 192.168.1.81/82 及 vCenter 同一个网段，并且确定能从 2003 这个 VM 上 ping 通网关，使用 ping –t 192.168.1.1。



▲ 开始 ping 虚拟机

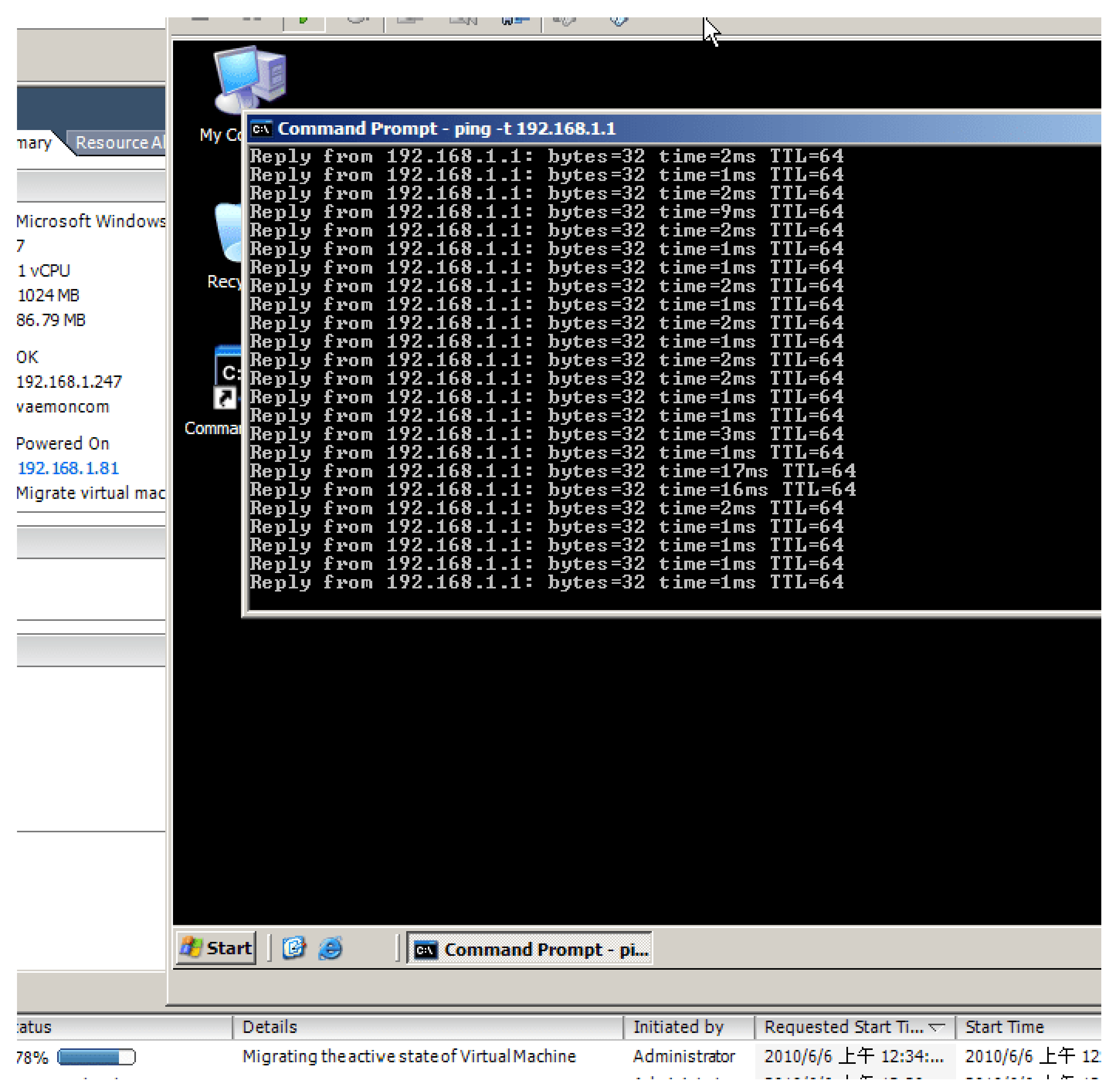
2．在 ping 的同时并发打开 VMotion，将 2003 这台虚拟机开始转移。

3．我们会发现本来 ping 的时间都在 1ms 以下，但开始 VMotion 之后，整个 ping 的速度慢下来，但还是通了。



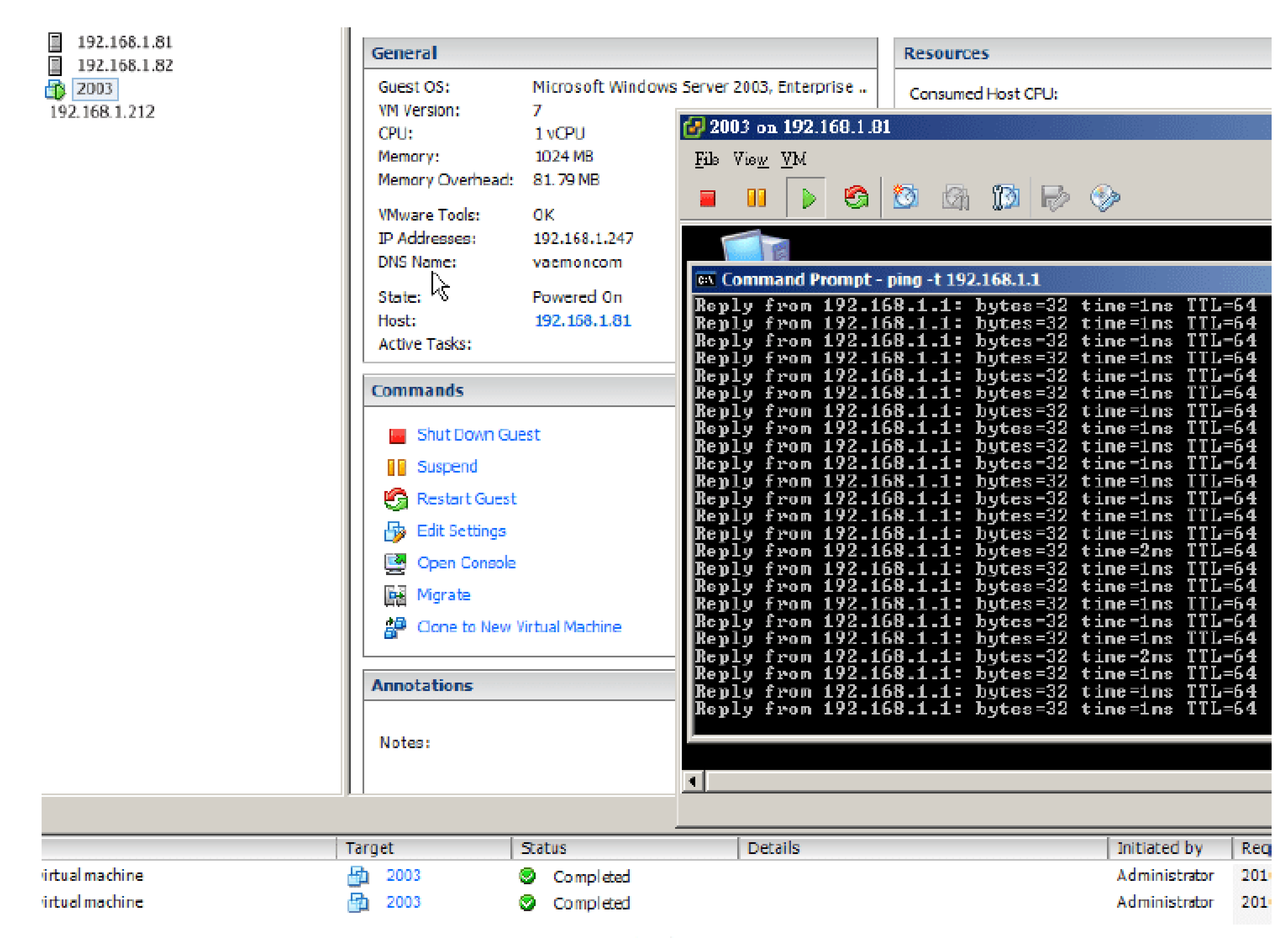
▲ 变慢的现象生成时，意味着开始转移了

4．此时还剩很短的时间，ping 速度在慢了一点之后立马恢复。



▲ 会短暂地停一下

5. 当 VMotion 落实时，ping 的速度恢复常规，显示整个 VMotion 时，虚拟机的服务几乎没有中断。



▲ 恢复常规

笔者在具体操作 VMotion 时，只要网络的速度够快，90%以上的是不会掉报文的，因此在 VMotion 的操作上，更快捷的网卡、交换机、网络存储设备等，会让整个 VMotion 的转移更快捷落实。

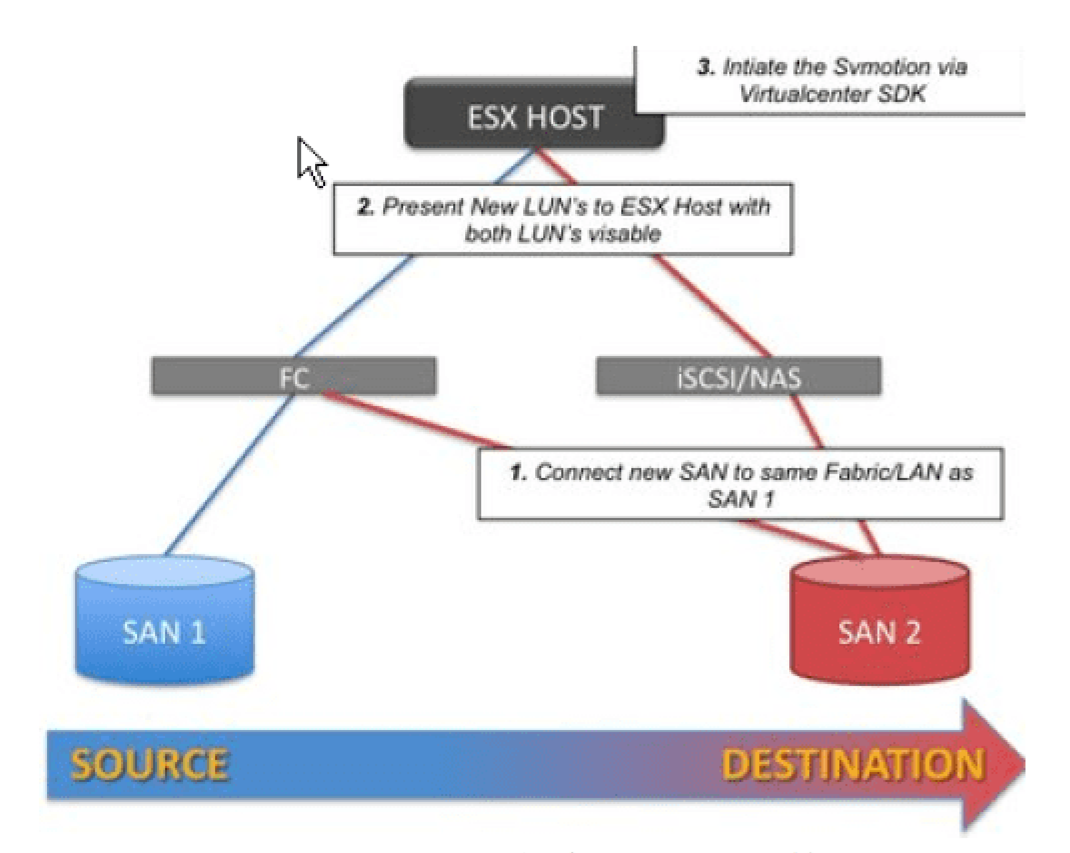
15.3 存储设备的 VMotion：SVMotion

虽然我们的生产环境是使用独立的存储设备，该存储设备也是使用 RAID1 或 RAID5 创建的，但难保这个独立的存储设备不会出疑义。此外在一个企业环境中，存储设备也会需要维备（如容量饱和或新建物理磁盘），因此 VM 的虚拟硬盘文件如果存储在独立的设备中，就算是 ESX 主机没有停机，当存储设备需要停机时，就也得将 VM 的磁盘搬移到其他的存储设备上，如本地或是其他的独立存储。

前面一直强调，企业最不想见到的就是服务中断，因此如果在搬移 Storage 中的虚拟硬盘时，必须将 VM 关机，那么再好的 HA 或 VMotion 功能也没用了。为此，VMware 在 vSphere 中也包括了一个功能，就是让存储设备中的虚拟硬盘也能搬移，这个功能就称为 Storage VMotion 简称为 SVMotion。

15.3.1 SVMotion 的原理

理解了 VMotion 的原理之后，要理解 SVMotion 就简单了。Storage VMotion 的原理并不需要有内存的克隆，就是很单纯的数据克隆，因此在 SVMotion 开始时，并没有主机的参与，因此只是单纯的文件在不同的存储设备间克隆，比起 VMotion 简单多了。



▲ SVMotion 的原理较简单，就是存储之间的转移

1. SVMotion 的基本条件

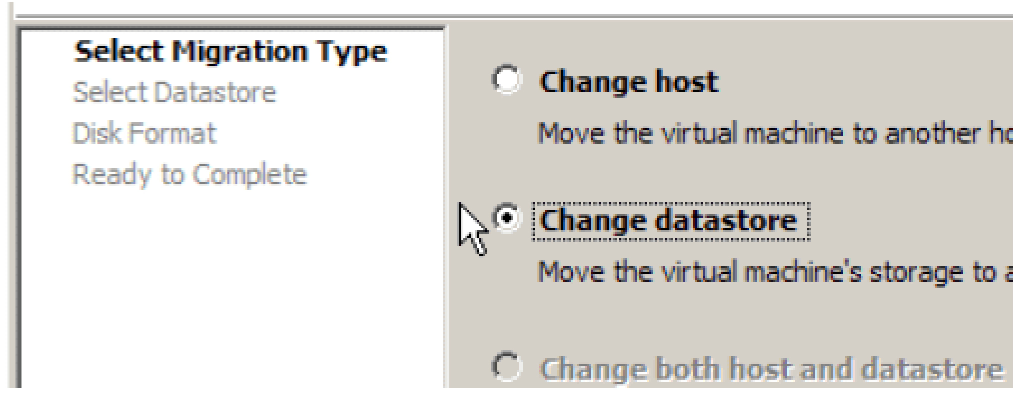
要支持 SVMotion，你必须支持下列的设备及条件。

▶ 支持 SVMotion 的条件

- 主机的 CPU 必须支持 VMotion。
- 要有两个存储设备，ESX 主机上的存储设备也可以。
- 两个存储有完整的网络连通。

2. vSphere SVMotion 的实践

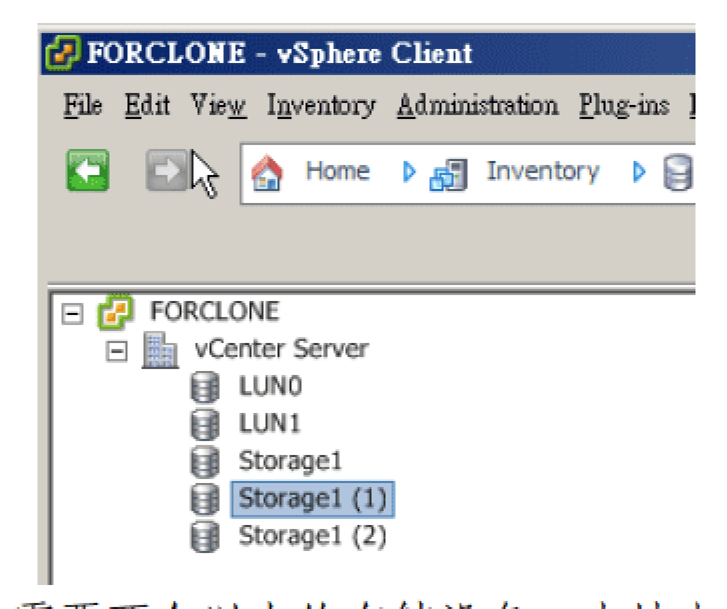
从前在 VI3 时，SVMotion 必须安装一个 Plugin 才能落实，但在 vSphere 中，已经将 SVMotion 集成到 vCenter 的 VMotion 中了。因此我们可以直接在 VMotion 的画面中来实践 SVMotion。



▲ 当前已经集成到 VMotion 中了

15.3.2 具体操作 SVMotion

SVMotion 的单纯创建在于整个 vCenter 环境的设计优良，本章的环境已经足以使用 SVMotion 了。在开始之前，别忘了将另一个存储设备接上你的两个 ESX 主机上，如此才能进行 SVMotion。接下来我们就从最简单的关机迁移开始。



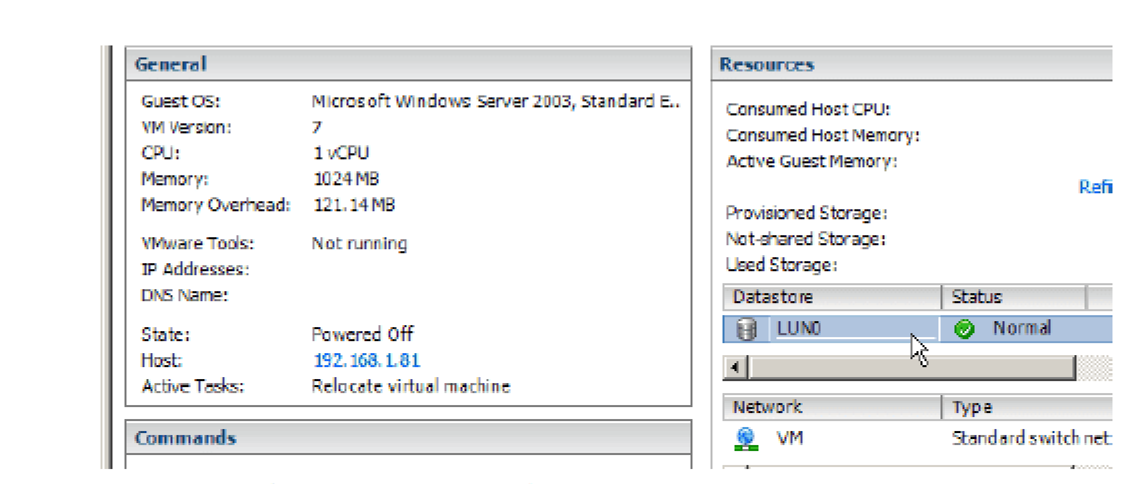
▲ 需要两个以上的存储设备，本地也可以

1. 实践关机状态的 SVMotion

事实上，在关机状态的 SVMotion 选项反而比较多，可以并发将 VM 和存储都搬到另一个存储设备，只是要注意的是，SVMotion 的搬运是完整的虚拟磁盘，因此花的时间会较久。

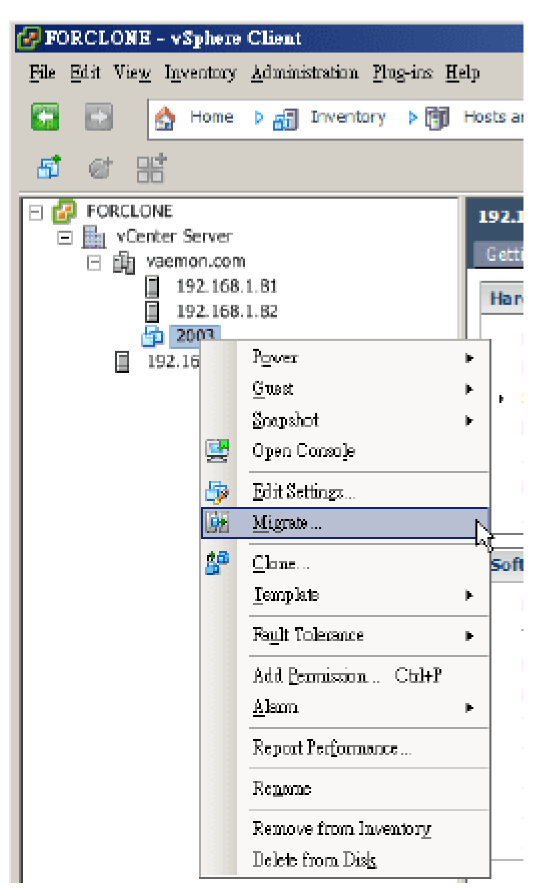
▶ 关机状态下使用 SVMotion

1. 选定拥有 VM 的 ESX 主机，并且确定其存储设备有两个以上。
2. 一开始的主机是在 192.168.1.82，存储是在 LUN0。



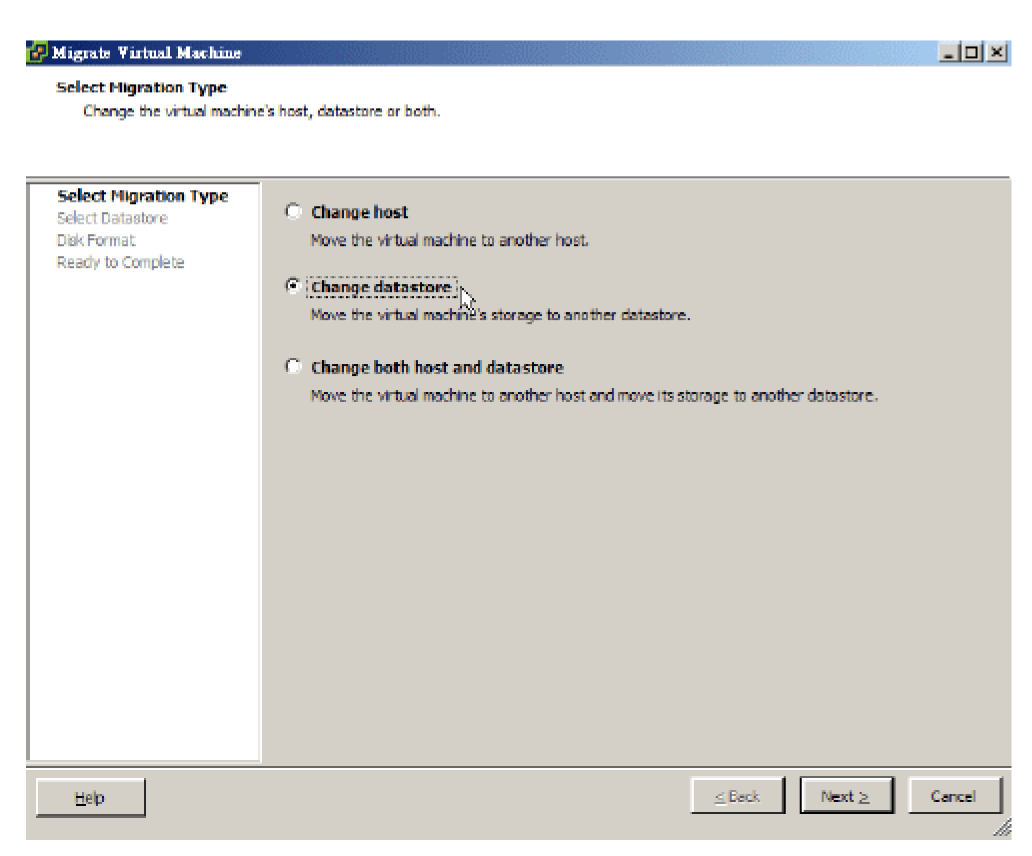
▲ 一开始的 VM 是放在 LUN0 上

3. 在要进行 SVMotion 的 VM 上右击，选择 Migrate 选项。



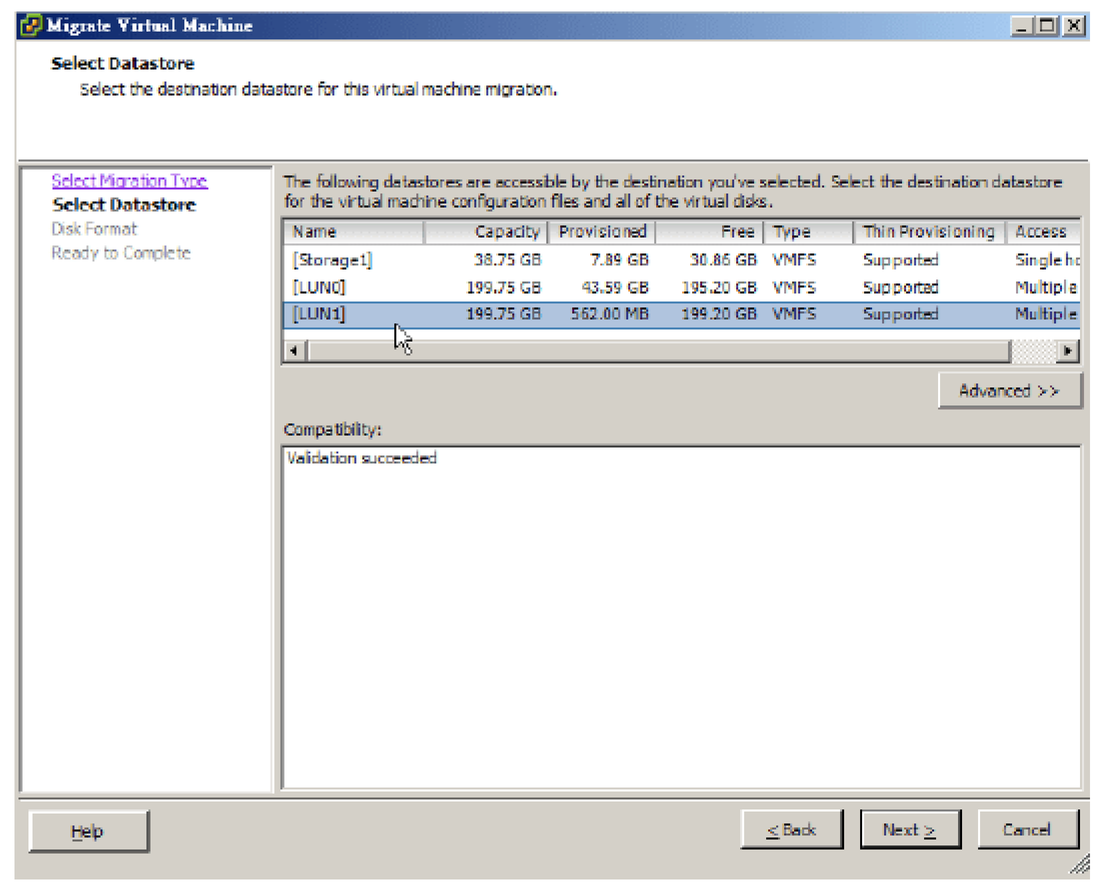
▲ 这里和 VMotion 的操作一样

4. 此时会有三个选项，第一个就是前面实践过的 VMotion，而第二个选项就是标准的 SVMotion。第三个则是将主机和存储设备都搬动。我们先选择第二项。



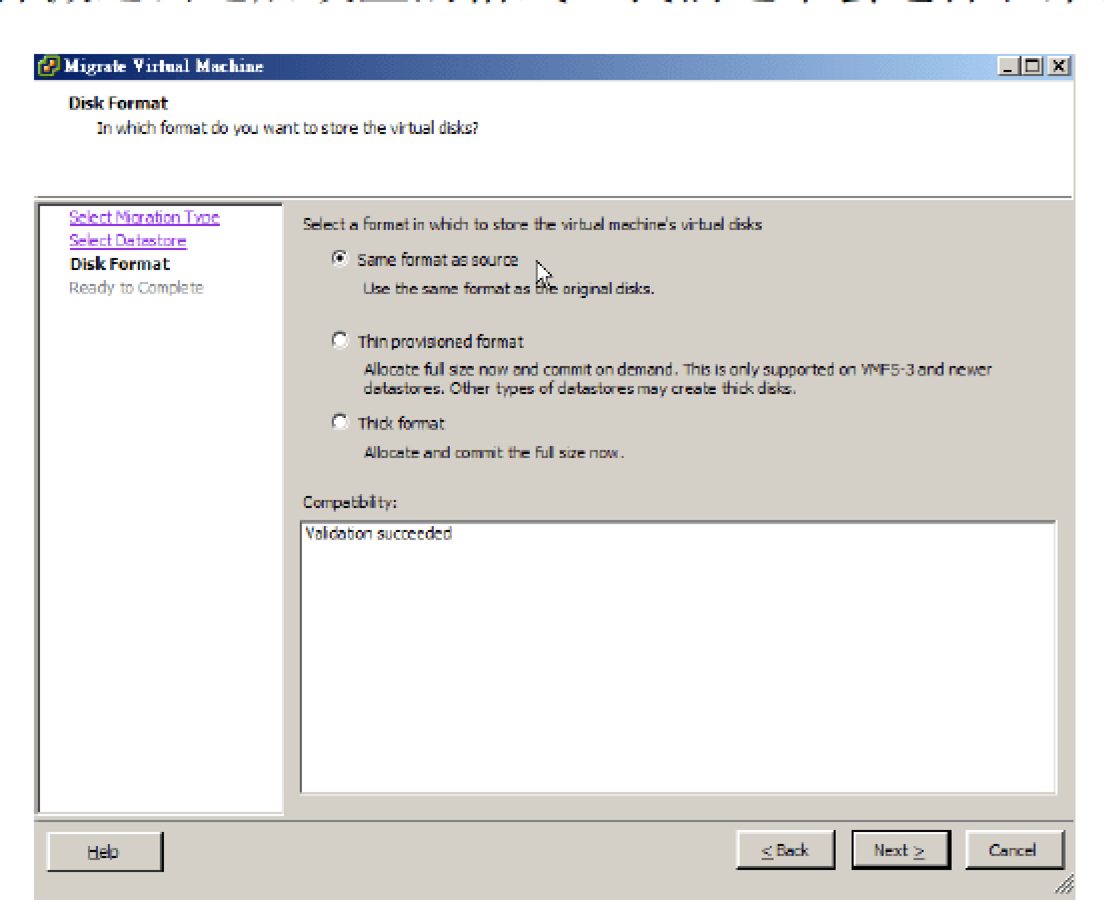
▲ 选择第二项

5．接下来选择要置放的新存储设备地点。如果这个地点可以的话，会弹出操作成功的消息。



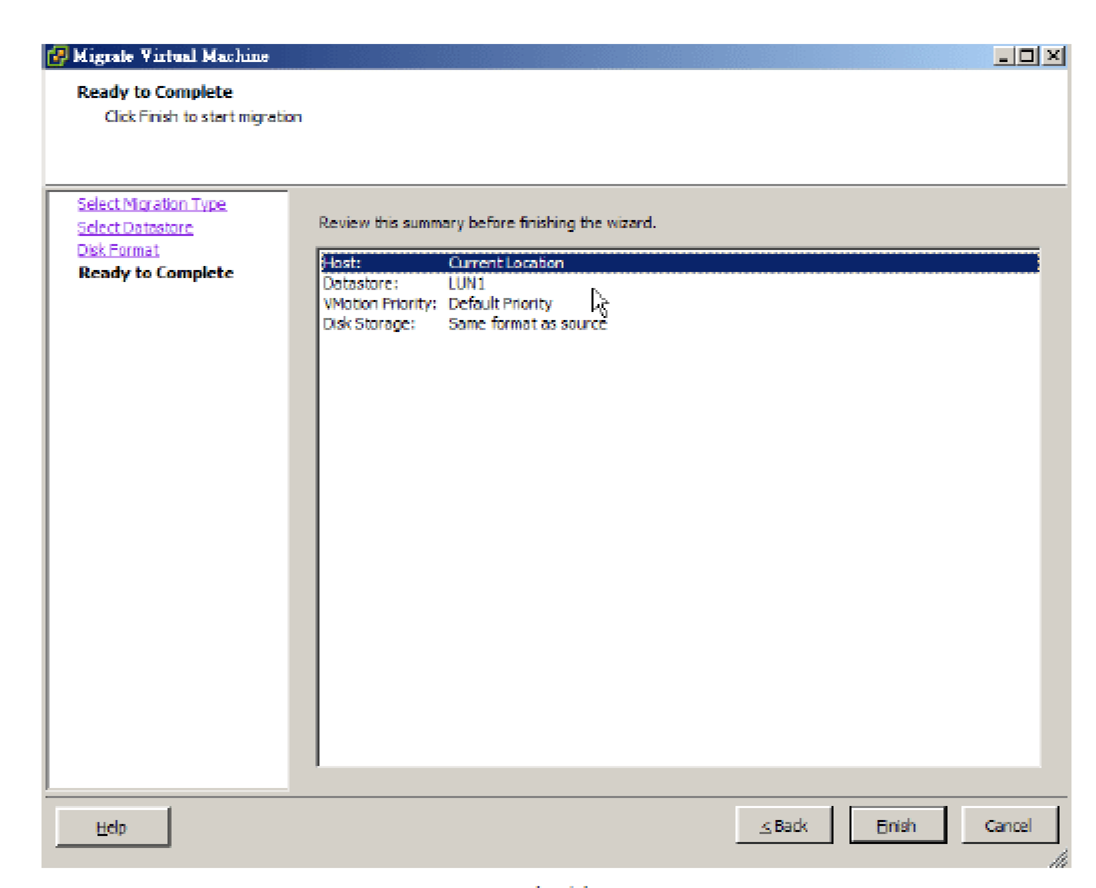
▲ 选择要放置的地点

6．接下来是选择搬过去之后硬盘的格式，我们通常会选择和来源硬盘一样。



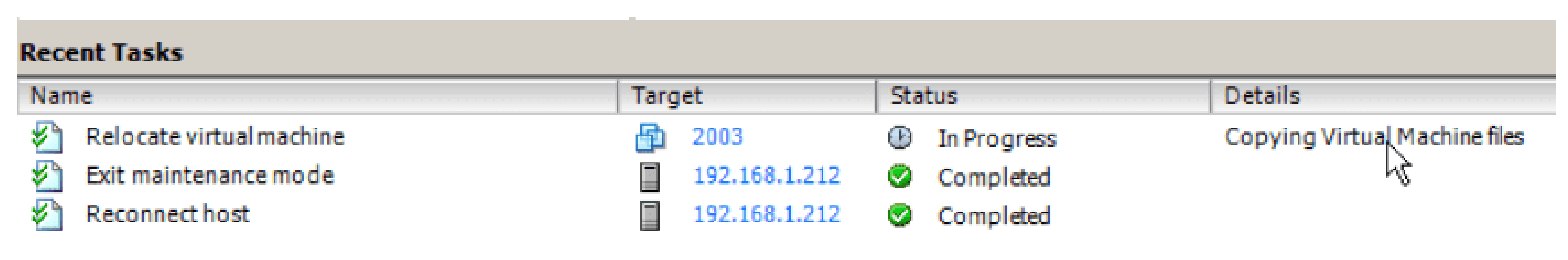
▲ 搬动后的硬盘格式

7．接下来会弹出总结画面，单击 Finish 按钮后就会开始动作。



▲ 总结画面

8．接下来就开始搬动了。



▲ 开始搬动

搬动完之后，会发现已经搬到 LUN1 去了。

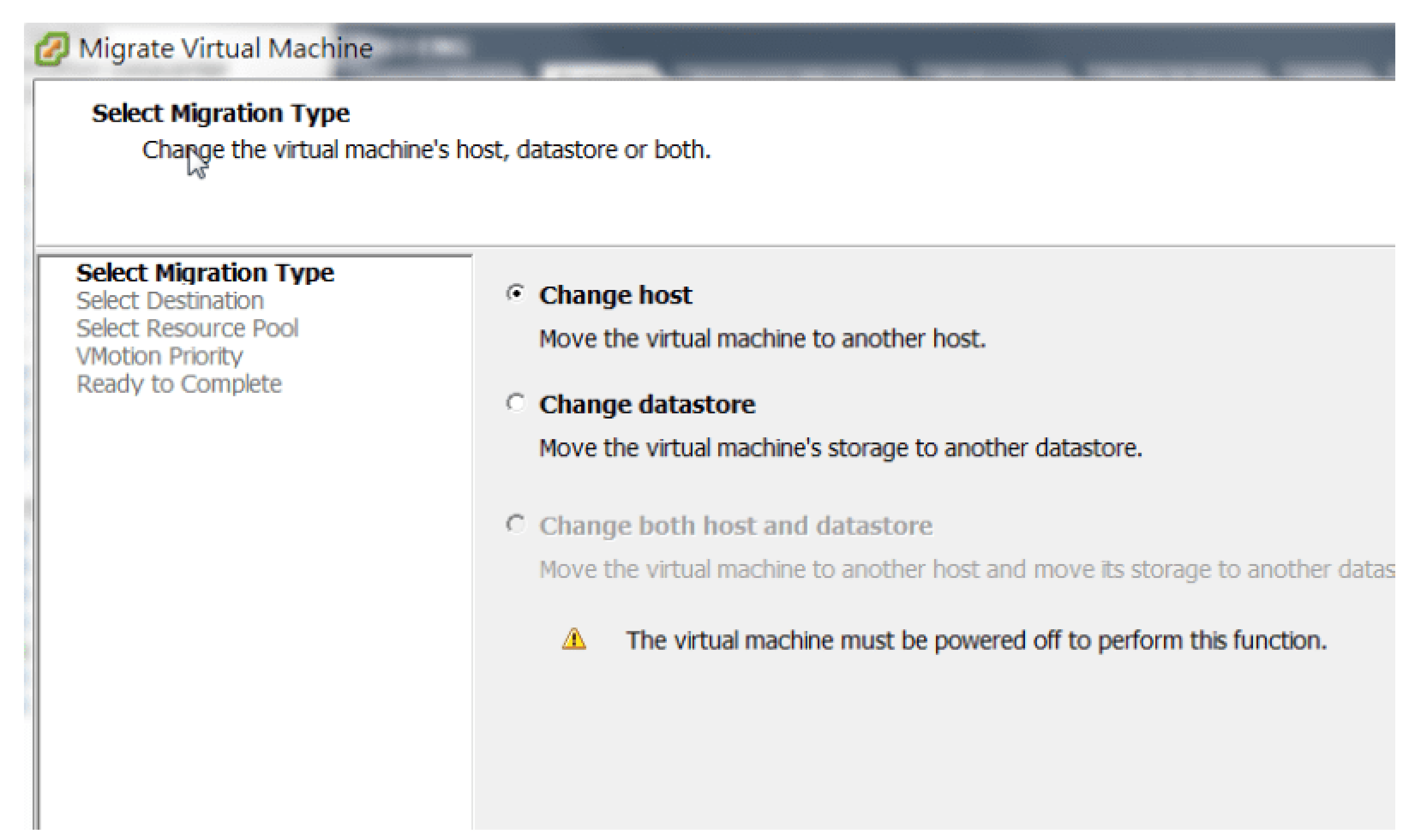
2. 引导状态的 SVMotion

引导状态的 SVMotion 也和 VMotion 一样，必须先将内存的状态复制，但由于整个操作只限定于一台主机，因此虽然速度慢，但主机也不会有任何停摆，下面我们就来看看完整的操作。

▶ VM 引导状态的 SVMotion

1．在 VM 的引导状态下，在该 VM 上右击，选择 Migrate 选项。

2．此时系统会要求我们是要换主机还是换存储，选择第二项的换存储。在引导状态，第三项是无法运行的。



▲ 引导状态是无法并发搬动 VM 和存储的

3．接着选择要将新的虚拟硬盘置放的位置，我们就从 LUN01 放到 LUN02 上。

4．接下来是选择移动后硬盘的格式，就选择和原有硬盘一样。

5．接下来是总结画面，单击 Finish 按钮就会开始移动。

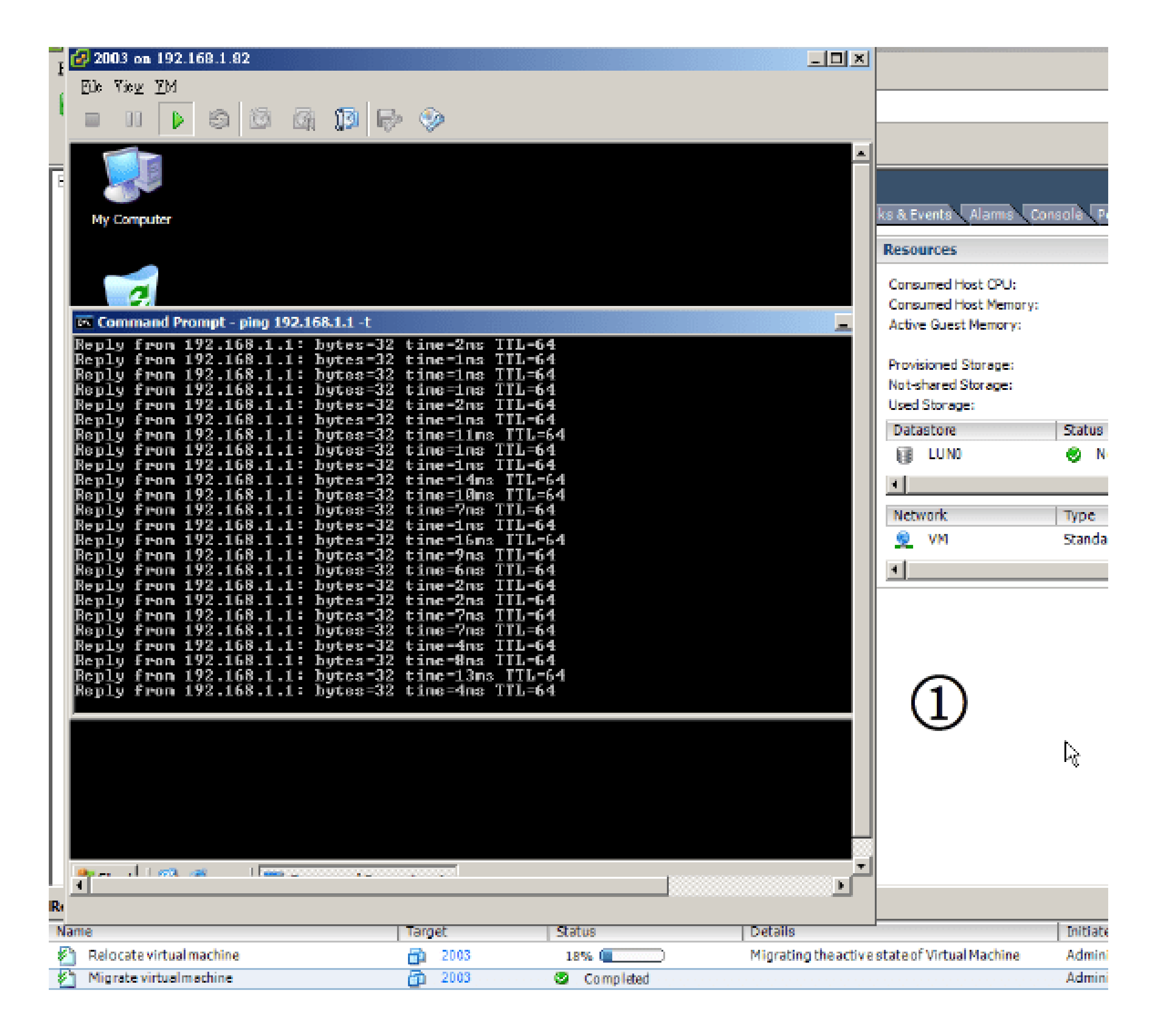
我们可以看到开始移动 VM 的硬盘了。

3. 近距离观察 SVMotion

我们也从 2003 的 VM 中来 ping 网络网关，并且直接观察 SVMotion 的作用情况，看看在整个过程中，虚拟机 2003 是否有停机或是无法使用。

▶ 观察 SVMotion 运行时虚拟机的状况

1. 首先我们把虚拟机的 IP 设成 192.168.1.X 的网段，并且确定能从 VM 上 ping 通网关，使用 ping –t 192.168.1.。

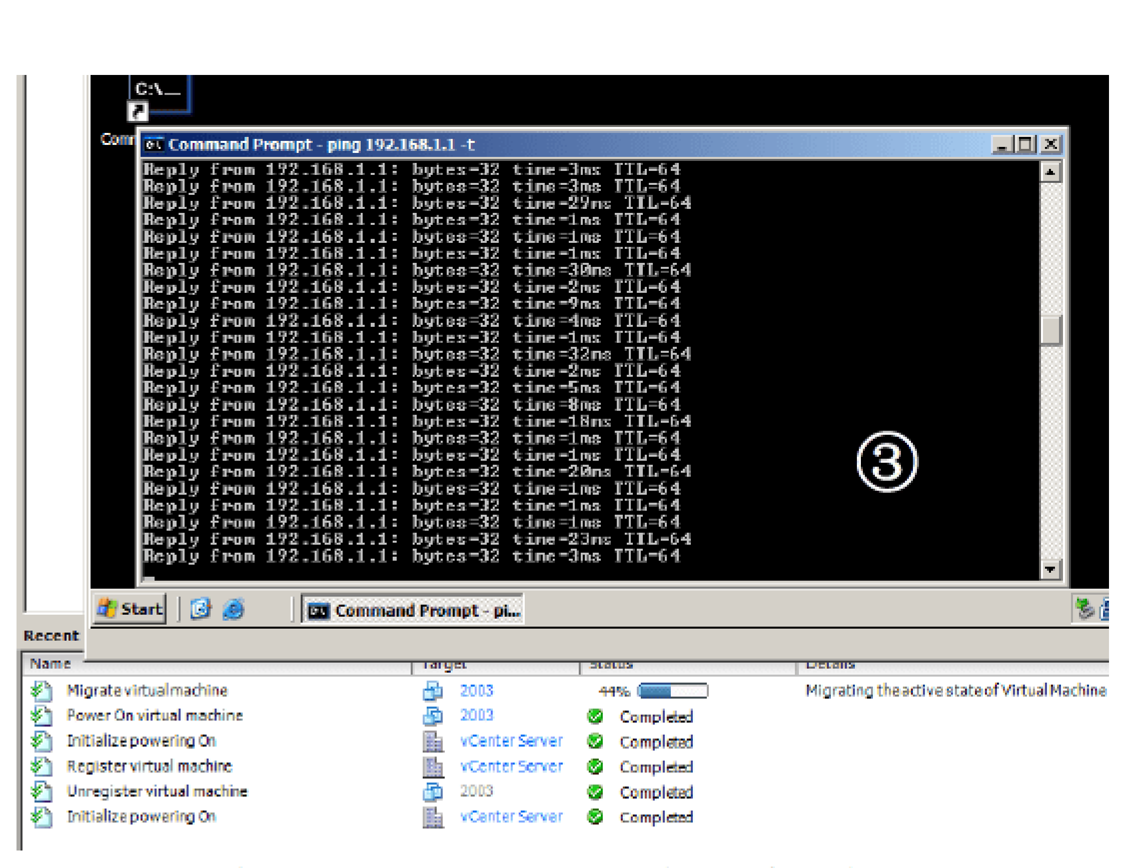


▲ 开始 ping 虚拟机

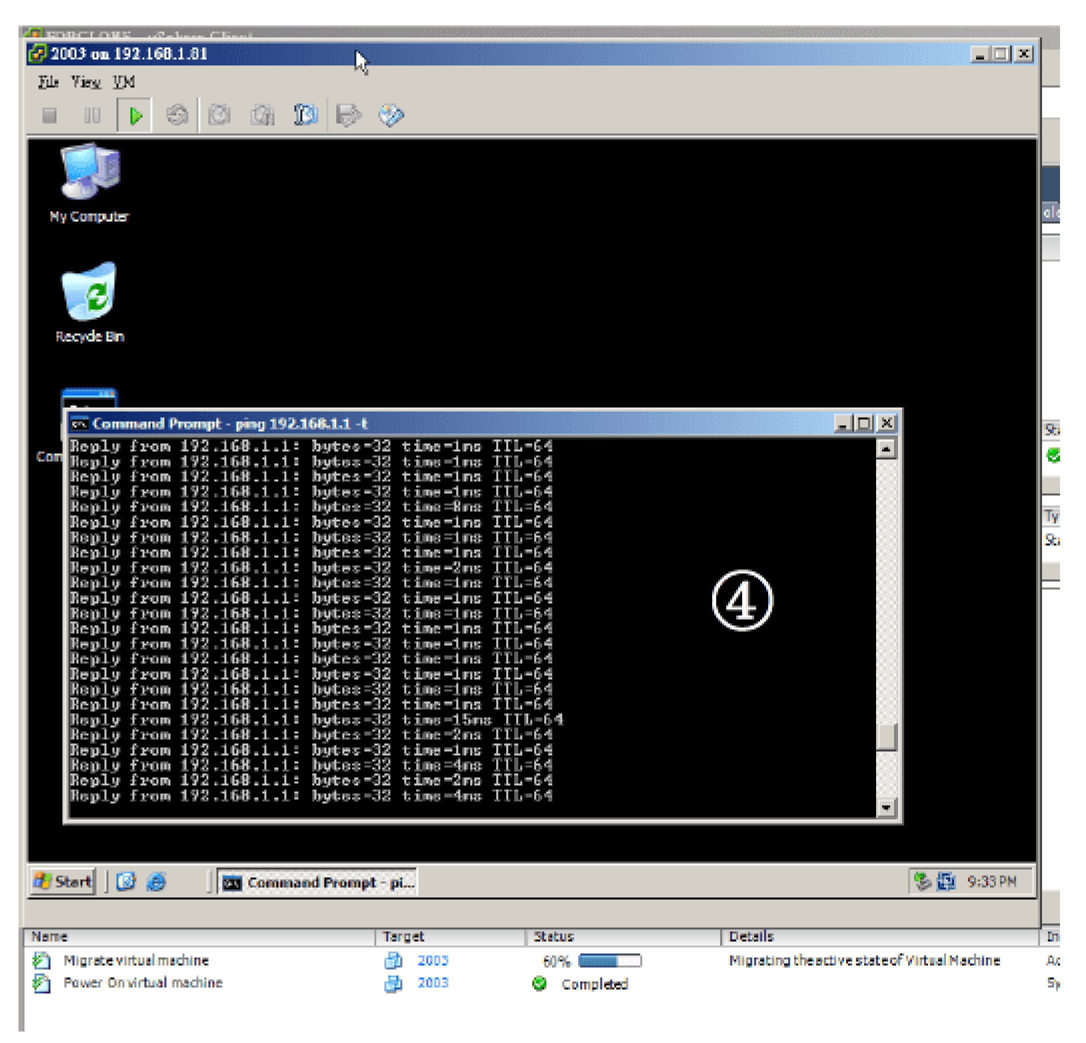
2．在 ping 的同时并发打开 SVMotion，将 2003 这台虚拟机开始转移。

3．我们会发现本来 ping 的时间都在 1ms 以下，但开始 VMotion 之后，整个 ping 的速度慢下来，但还是不通了。

4．在 vSphere 下，竟然一个报文都不会掉！



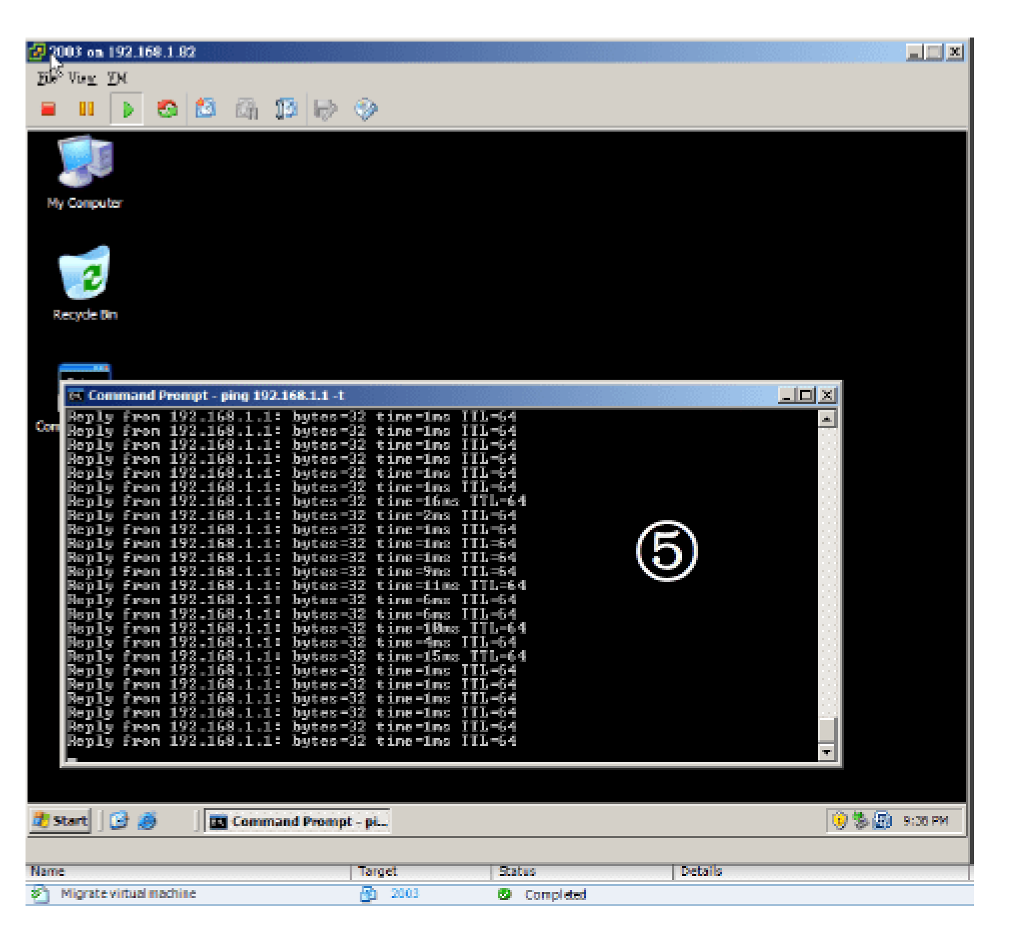
▲ 变慢的现象生成时，意味着开始转移了



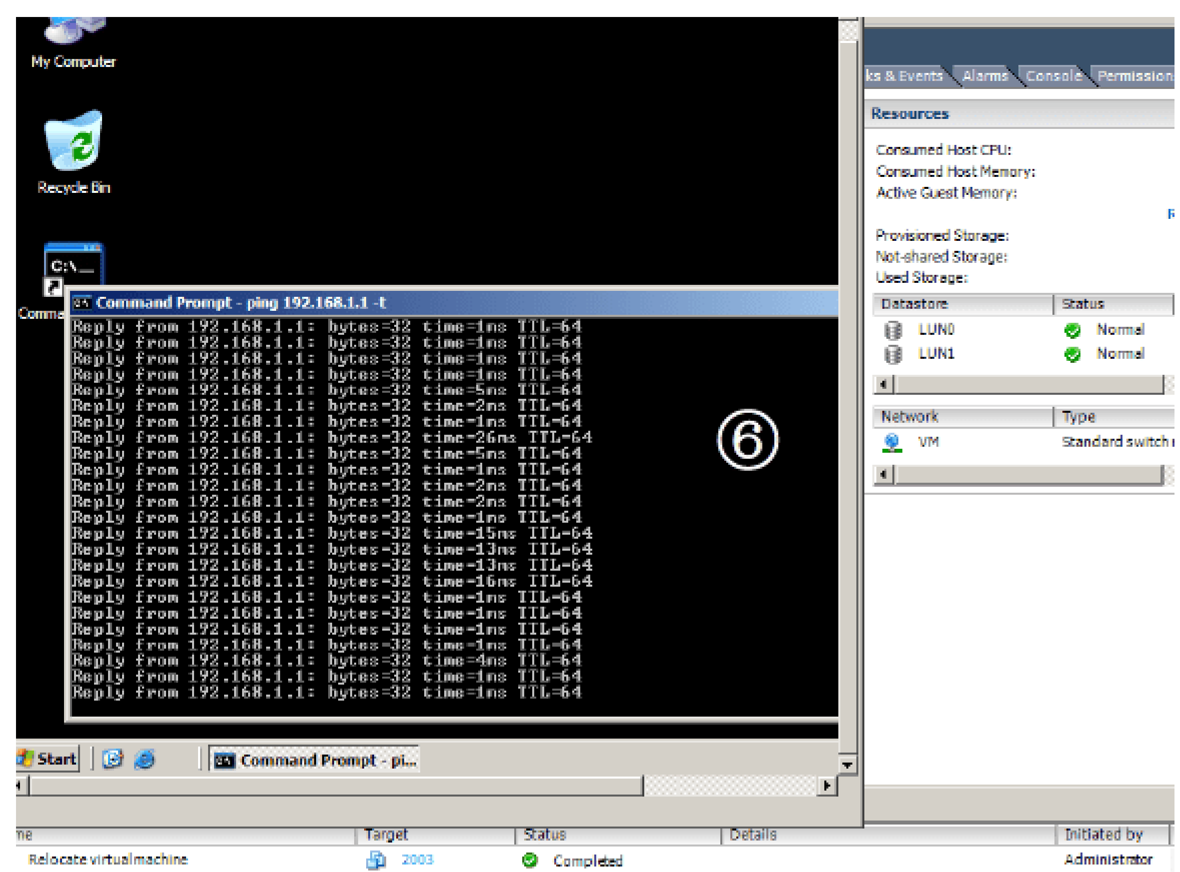
▲ 大部分的情况下并不会掉报文

5．当 SVMotion 落实时，ping 的速度恢复常规，显示整个 VMotion 时，虚拟机的服务几乎没有中断。

6．要注意的是，VMotion 会将 VM 搬移，但 SVMotion 会将 VM 的 VMDK 克隆，因此在 SVMotion 之后，在 LUN0 和 LUN1 上各会有一份 2003 这个 VM，因此要注意版本的问题。



▲　恢复常规



▲　会在两个存储上都留下 VM，SVMotion 的目的是副本

结　语

VMotion 是一切高级 VI 架构的基础，在 ESX 主机之间移动虚拟机的 DRS、HA 架构都靠着 VMotion 来进行。也许读者们在经过了将近 90%的设置，发现 VMotion 竟然只需要几个动作就可以落实，但 VI 架构的庞大和复杂，必须有多方的配合，包括硬件、软件、Active Directory 域、数据库、vCenter Server、网络环境、独立存储设备等。vSphere 将 SVMotion 也简化不再需要 Plugin，使得存储设备的转移也变得简单，而有了 VMotion 和 SM 的双重保障，vSphere 环境的安全性也大幅提高了。

读书笔记

第 16 章

vSphere 的分布式资源调度功能：DRS

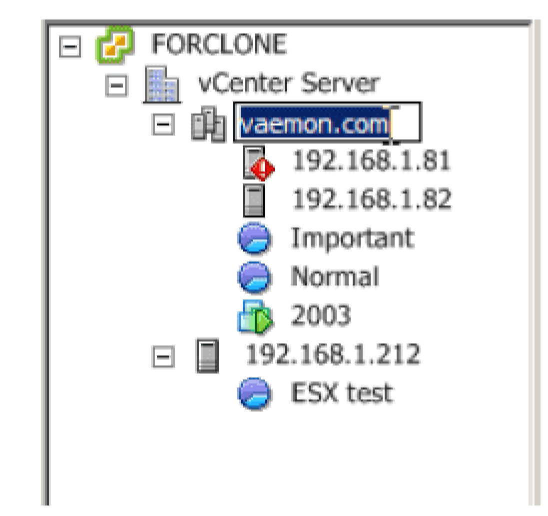
关键词：

- vSphere 中的资源池
- 创建 vSphere 中的集群
- DRS 的运作原理
- 全自动 DRS 的五个选项
- DRS 的规则配置（Rules）
- 在 vCenter Server 中实践 DRS

我们在第 15 章中曾经提到，VMotion 不但是一个 vSphere 中的功能，更是其他更先进功能的基础。vSphere 支持分布式资源调度，系统会在最需要的时候将不同的虚拟机在不同的 ESX 主机上激活，vSphere 也支持让虚拟机永远不会在任何 ESX 主机宕机时跟着一起停止服务。因此在成功落实 VMotion 之后，我们就在这一章中，来看看这个最重要的 vSphere DRS 功能，这也是 vSphere 最重要的价值所在。

16.1 理解 vSphere 的集群（Cluster）

多台电脑组成的集合就称之为集群，集群的目的就是将计算机的负担分配到多台主机上，或是当运行某一个服务的物理服务器弹出问题时，必须让另一台服务器接手这些服务。在 vSphere 上，这些服务都应用在一个个的虚拟机中，因此能让多台物理服务器将其上的虚拟机在需要时，在不同的服务器之间移动，就是 vSphere 中集群的主要功能了。

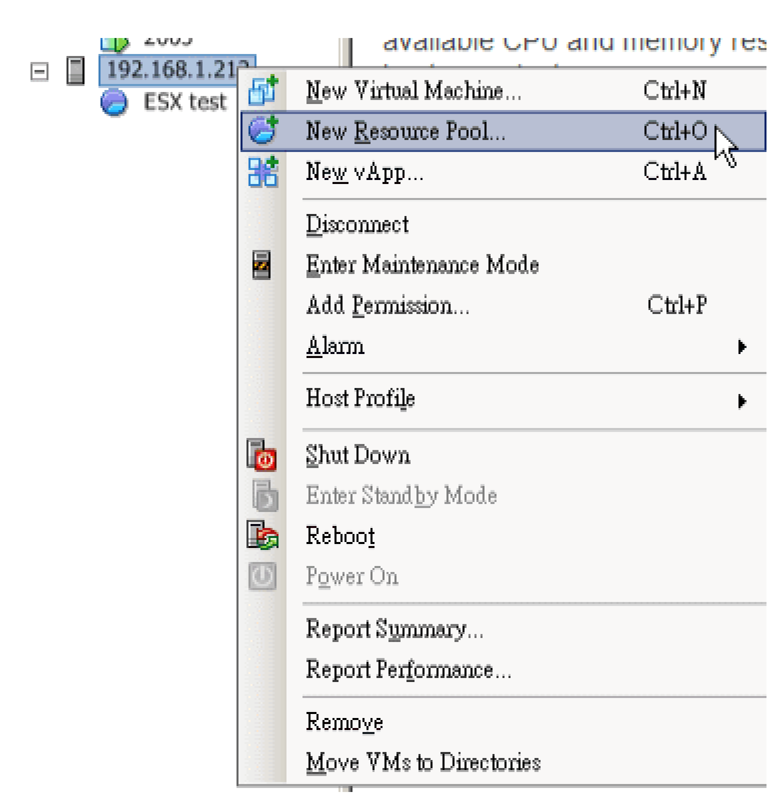


▲ 在 vSphere 中，所有的 DRS 和 HA 操作的基本单位都是以集群为主

举例来说，某台 ESX 物理服务器上运行了 10 个虚拟机，其中有一台是运行 IIS 作为网页服务器。当这台网页服务器的用户越来越多的时候，物理 ESX 服务器上的其他虚拟机将会因为资源不够而效能降低，此时如果我们能将这台运行 IIS 的虚拟机移到其他资源较充分的物理 ESX 服务器上，不但系统资源获取缓解，IIS 的使用和其他服务都能恢复常规，更在利用 VMotion 之后，所有的服务在过程中都不会中断，因此集群在 vSphere 中的重要性可见一斑。

16.1.1 vSphere 中的资源池

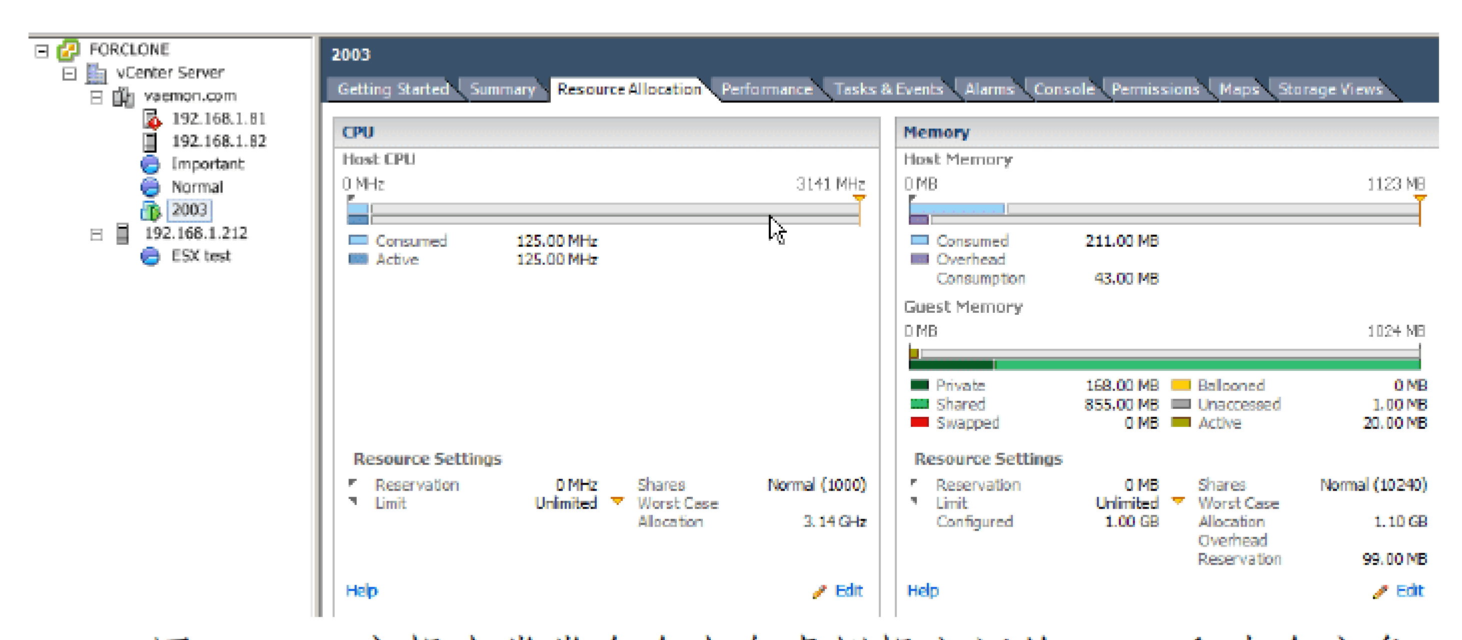
vSphere 的集群是由 2～32 台物理的 ESX 服务器组成，其能提供虚拟机在上面转移的高级功能，如分布式资源调度（Distributed Resources Scheduler，DRS）以及高可用性（High Availability，HA）等服务。再加上 VMotion 的操作之后，这些 ESX 服务器组成的架构可以让虚拟机在不同的 ESX 物理机上动态移动，从而落实永不中断的系统服务。



▲ 单个 ESX 主机中就已经有资源池的概念了

1. 对 CPU 和内存的竞争

在 vSphere 中，虚拟机是否能顺利运行，主要就是看其使用的 CPU 和内存是否足够。在物理机环境下，CPU 和内存不够只要添加内存或是将系统升级，通常是负担得起的，但在虚拟机环境下，如果这是一个规划完善的虚拟机架构，那么发生 CPU 或内存不够的情况，通常是来自虚拟机之间的竞争。因此在 vSphere 架构下，需要有一个机制协调这个竞争，这个机制就是资源池。

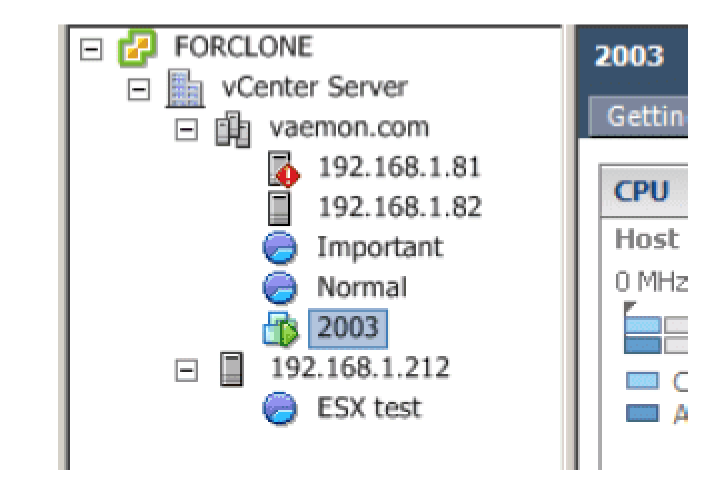


▲ 同一 ESX 主机上常常会有本身虚拟机之间的 CPU 和内存竞争

2. 集群中的重要部件：资源池（Recourse Pool）

资源池是 vSphere 中最重要的观念，主要还是用于虚拟机之间对资源的竞争。在一台物理 ESX 服务器上，虚拟机最主要的竞争就是 CPU 的时钟和内存。如果 ESX 物理服务器上的资源足够虚拟机使用（如 4 内核的 CPU 两颗上跑 16 个虚拟机），那么永远不会有资源的竞争（Competition）。但当资源不够时，就会生成虚拟机竞争，而竞争的协调者，就必须靠资源池了。

资源池是 vSphere 架构下的一个对象，主要是规定在虚拟机生成竞争时，如何分配 CPU 和内存。因此在资源池中的虚拟机，都必须遵守这个资源池的规则。举例来说，如果我们规定一个资源池中所有的虚拟机都不能使用超过 1000MHz 的 CPU 和 1024MB 的内存，只要你将任何虚拟机放入这个资源池，这个虚拟机就会遵守这个资源池中的配置，因此虚拟机的效能也会在生成竞争时有所变化。



▲ 在 ESX 或集群上可以创建资源池来合理分配 CPU 和内存

3. 资源池的重要部件

资源池的存放地点有两种，一个是物理的 ESX 主机，另一个就是 Cluster 了。是以 ESX 物理机为主，创建好资源池之后，这台 ESX 服务器上的虚拟机就可以用拖拉的方式移到资源池。虚拟机进入此资源池之后，就必须遵守上面的规则了。在 vSphere 中，资源池的规则主要分为 CPU 和内存，而这两大部分中也有一些细部选项，我们就来看看。

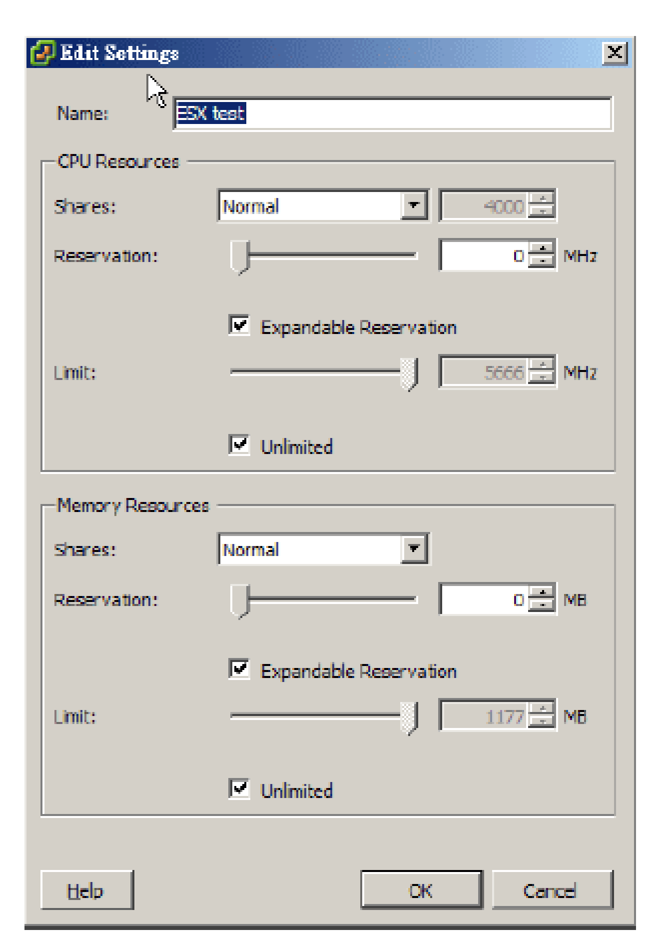
▶ 资源池的重要部件

CPU 部分：

- Share：一共有 Low（1600）、Normal（4000）、High（8000）及自定义等。这里主要是分配 CPU 的共享层级。越高当然使用的 CPU 越快。
- Reservation：保留使用的配置。这里是以 CPU 的总时钟来划分的。如一个系统有 2 颗双内核 3GHz 的 CPU，那么总时钟就是 2×2×3000 = 11600MHz。由此来划分。
- Expandable：当虚拟机需要短时间而大量的 CPU 资源时，可以将这个地方调高。举例来说，AD 域的域总控器在早上大家登录域时，就会需要较多的资源，但平常资源较少，就可以在这里配置。
- Limit：CPU 保留资源的上限。
- Unlimited：选中此项就没有上限。

内存部分：

- Share：一共有 Low（81916）、Normal（163840）、High（327680）及自定义等。这里主要是分配内存的共享层级。越高占用的内存越多。
- Reservation：保留给这个资源池的总内存，以 MB 为单位。
- Expandable：当虚拟机需要短时间而大量的内存资源时，可以将这个地方调高。
- Limit：内存保留的上限。
- Unlimited：选中此项就没有上限。



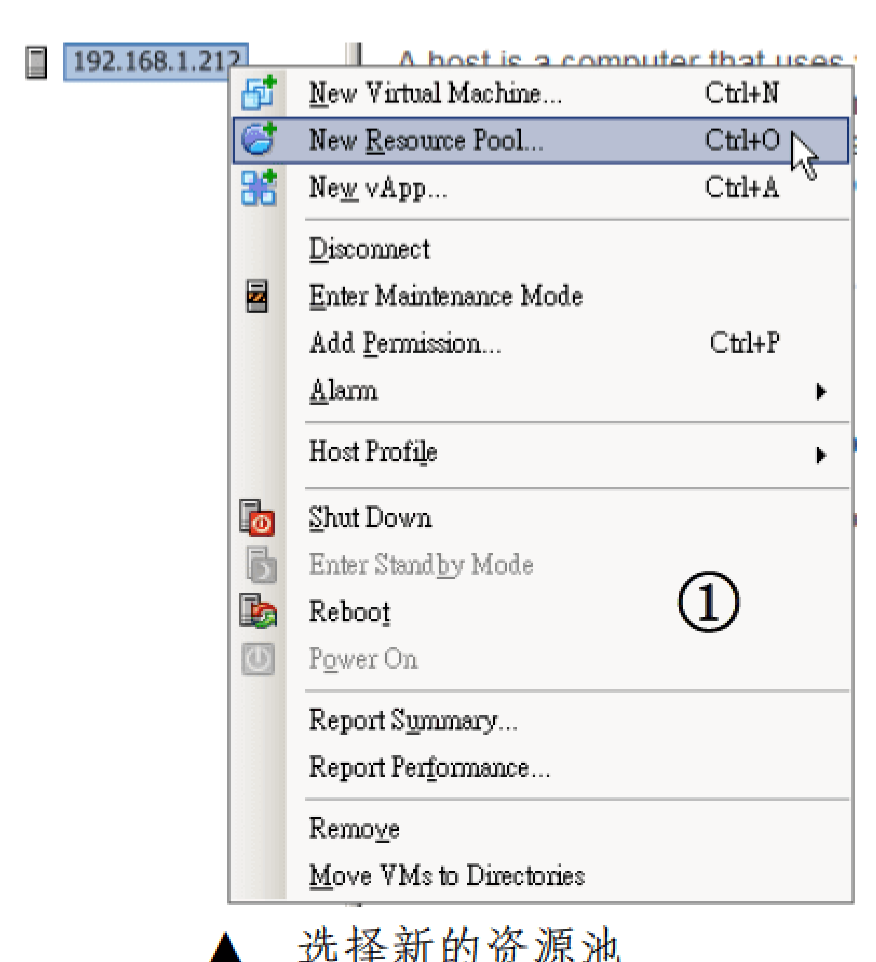
▲ 内存和 CPU 的分配是资源池的最重要任务

4. 在 vCenter Server 中替 ESX 新建资源池

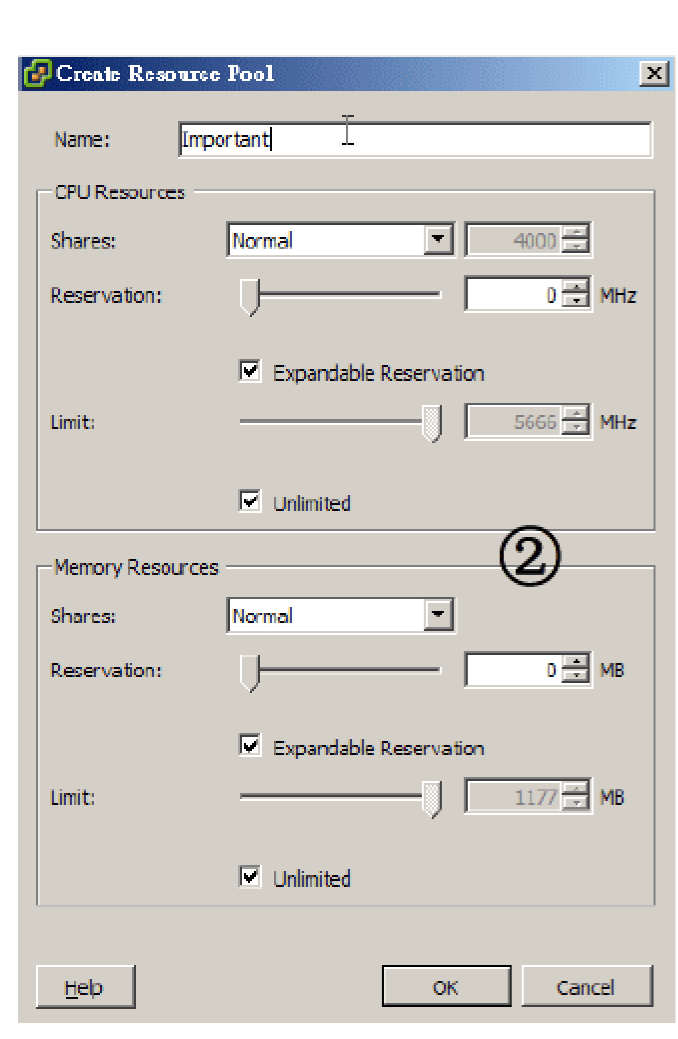
举例来说，如果我们有一台 ESX 服务器，上面运行许多虚拟机，但有些虚拟机不需要太多的资源，另外一些则需要较多的资源，我们就可创建两个资源池，一个为 Important，另一个为 normal，并且将这两个资源池中对于 CPU 和内存的配置分别来进行调整，然后再将这台 ESX（或在 vCenter Server 中其他 ESX）上的虚拟机拖入资源池，就可以落实资源池的分配了。我们就具体操作一次。

▶ 创建及配置资源池

1. 在如下图所示的 192.168.1.212 上右击，选择 New Resource Pool 选项。
2. 首先我们在名称上键入 Important。接下来就可以配置 CPU 和内存的资源池，这里的资源都以 High 为主，并且保留 2000MHz 和 1GB 内存。



▲ 选择新的资源池



▲ 进入之后就可以进行分配了

3. 接下来我们也创建一个 Normal，都使用 Low，并且不保留，上限配置为 2GB。
4. 创建好之后，可以看到在 192.168.1.212 这台主机上已经有两个资源池了。
5. 如果我们希望原来在 192.168.1.81 上的虚拟机能拥有保留的资源，可以直接将 2003 拖

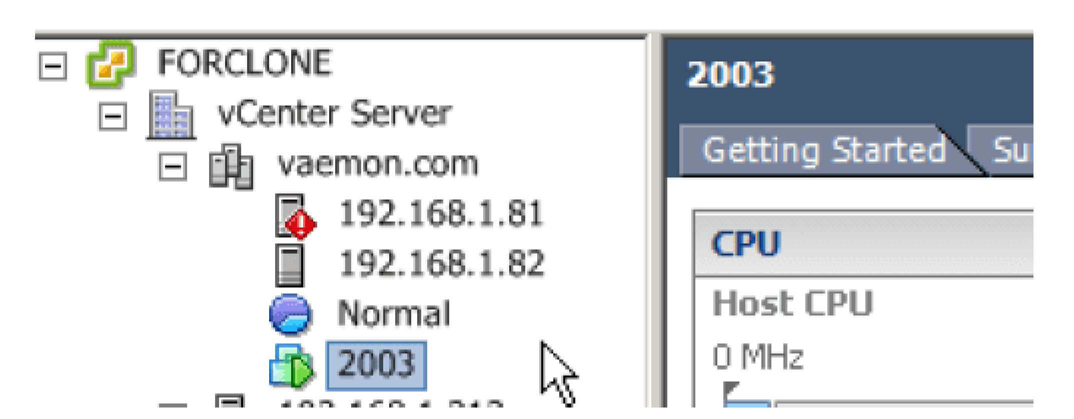
入 Important 中，反正我们也可以将 2003 这个虚拟机拖入 Normal 中。



▲ 两个资源池都创建好了　▲ 将虚拟机放入资源池的意义就是这个虚拟机必须遵守其分配规则

16.1.2　创建 vSphere 中的集群及资源池

在 vSphere 中，一个集群的创建相当简单，不需要特殊的硬件，只要参与集群的 ESX 主机能被 vCenter Server 所总控，并且连在同一个网段上，能看到彼此的主控台即可。此外 vSphere 要求所有的集群中的主机必须在同一个 Datacenter 中。我们接下来就来看看如何创建集群。



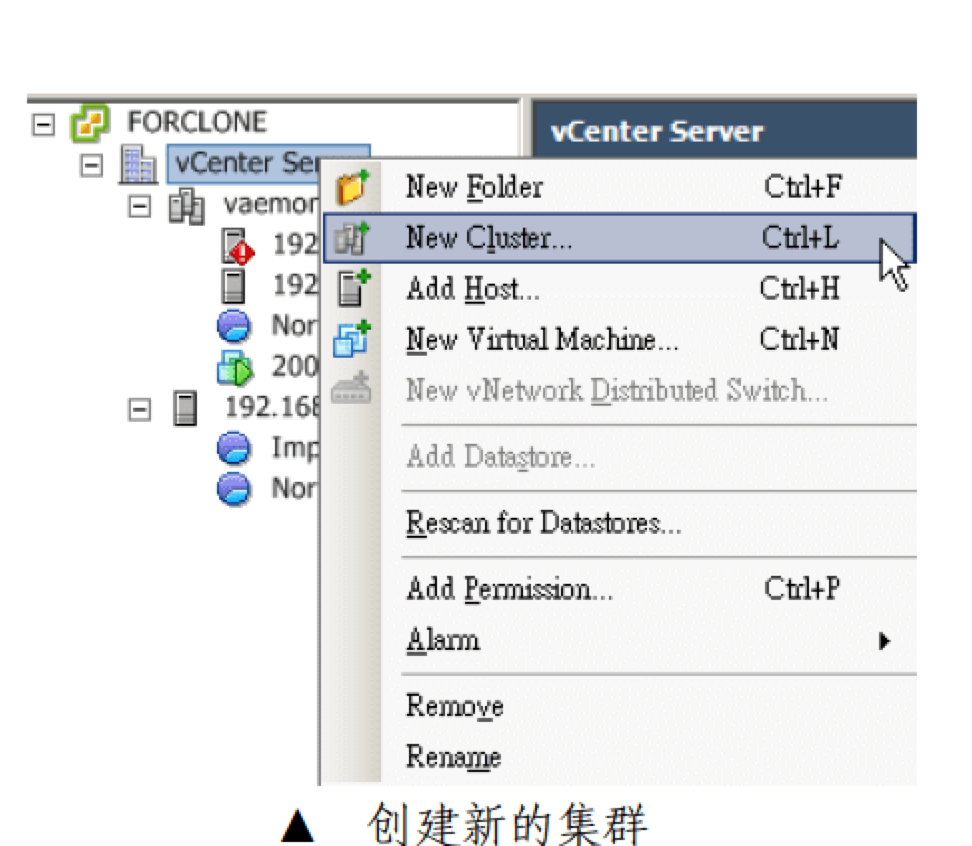
▲ 集群的部件包括两台以上 ESX 主机、虚拟机和资源池

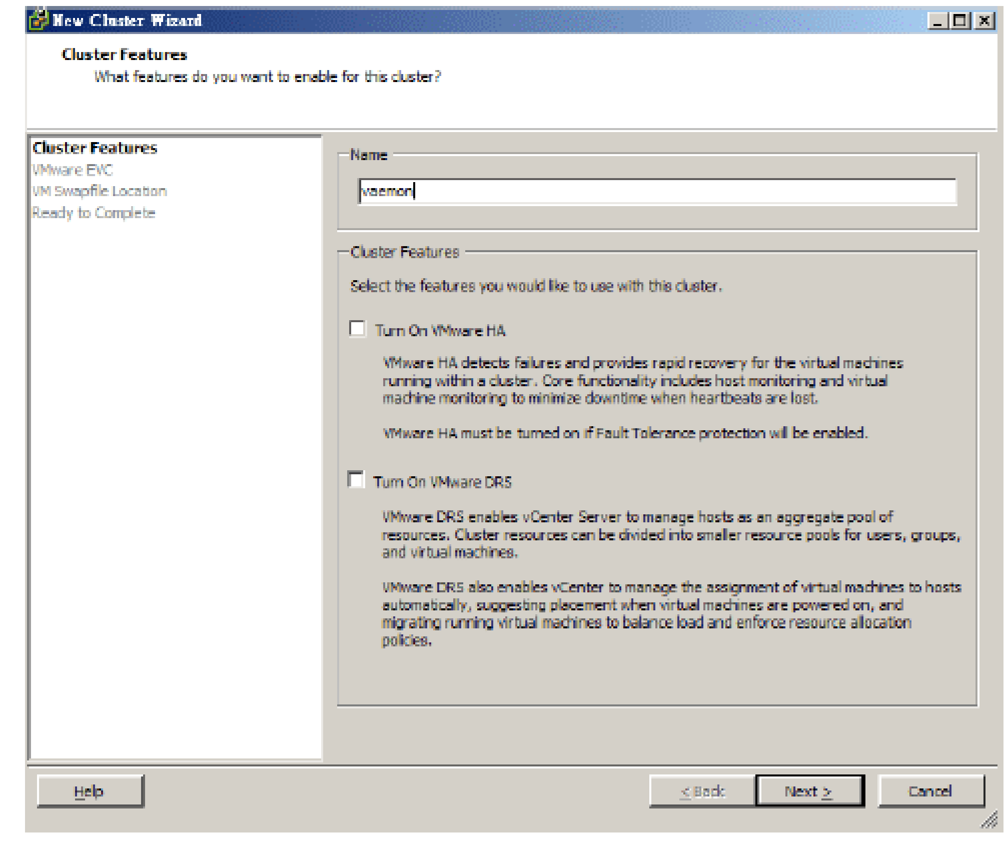
1. 创建标准的 vSphere 集群

在 vSphere 下创建集群十分简单，主要就是将需要的资源激活，并且将参与集群的主机加入即可。我们就先来看看如何创建标准的集群。

▶ 在 vSphere 下创建集群

1. 选择一个 Datacenter，并且在其上右击选择 New Cluster 选项。
2. 键入这个集群的名称，我们可以直接在这里激活 HA 和 DRS 功能，也可以之后再选中，我们就先不选中。

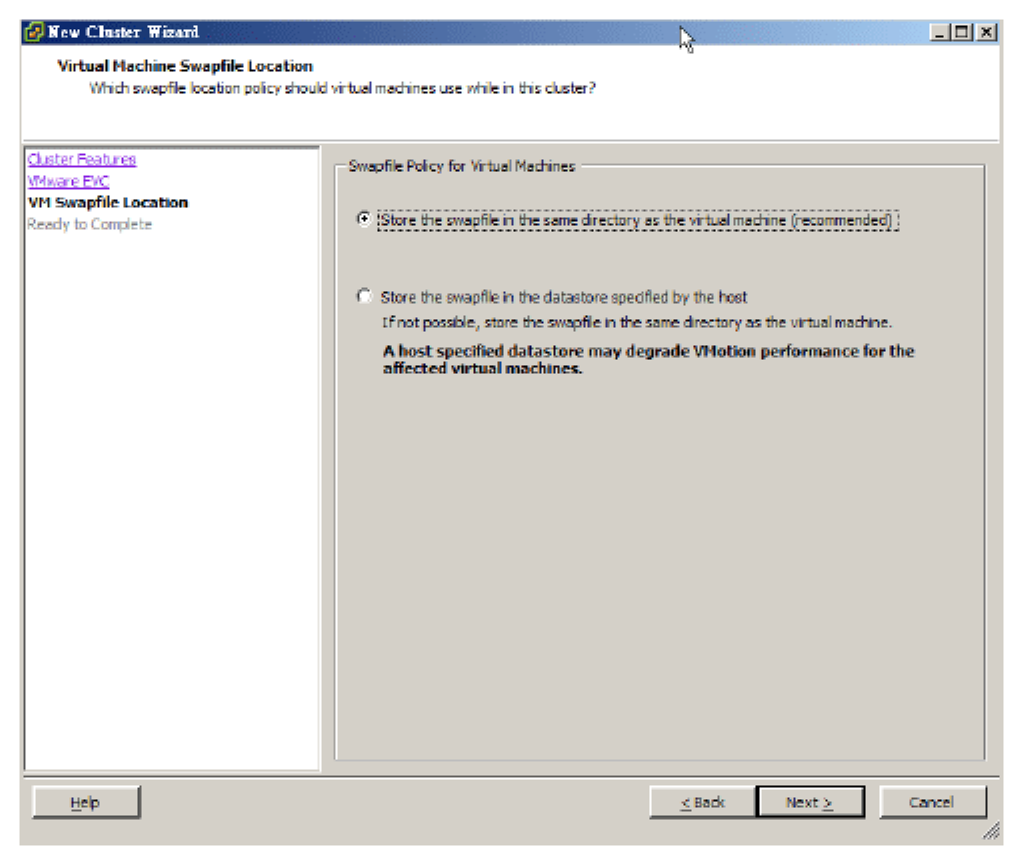




▲ 创建新的集群　▲ 可选择激活 DRS 和 HA 功能

3. 接下来是配置 VMotion 的选项，我们在下一章会讲到。接下来就是虚拟机的置换文件的存放位置。默认是存放在虚拟机 VMX 和 VMDK 的同一位置，也可以自行给定，但效能会变差，因此我们就使用默认值。

4. 接下来单击 Finish 按钮即可开始安装，当安装完毕后，可以在 vCenter Server 的主画面看到这个集群。

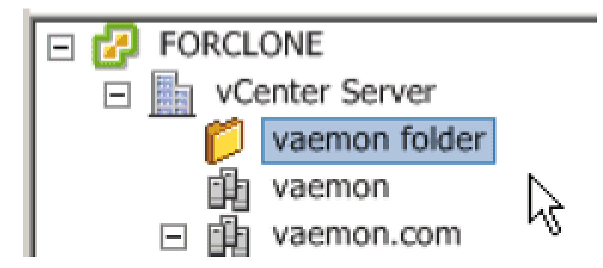


▲ 配置集群的存放位置

▲ 集群创建完毕了

2. 添加集群的主机

创建好集群之后，当然就要将 ESX 的主机加入。加入 ESX 主机很简单，只要将左边代表主机的图标直接拖入集群中，就可以看到主机加入了。当然一个集群是由两台以上的 ESX 主机所组成，因此我们要使用集群的功能，一定要有两台以上的主机才行。



▲ 不到两台主机就不能称为集群，而是文件夹 Folder

16.1.3 集群中的资源池

集群中的资源池比单个 ESX 主机上的还要重要，因为在集群中创建资源池之后，整个资源池管理的就是所有 Cluster 中的 ESX 主机资源了。举例来说，如果这个 Cluster 中有三台 ESX 主机，那么我们可以管理的资源就是这三台 ESX 的所有 CPU 和 RAM 的总和了。

> **注意**
>
> **单个 VM 使用资源池中的最大资源**
>
> 在 Cluster 中的资源池虽然能管到多台 ESX 物理服务器的 CPU 或内存的总和，但要注意的是，单个 VM 能用到的最大 CPU 及内存，还是在制约单个的物理 ESX 服务器的资源。举例来说，你有两台 ESX，分别使用两个四内核的 3GHz CPU，那么一台 ESX 就有 8 个 3GHz 的 CPU，一共有 24000MHz，两台就有 48000MHz。但你在分配给单个 VM 时，最大还是只能留 24000MHz，就算你在资源池中分配超过 24000MHz 也没有用，因此 VM 是无法并发访问多台物理 ESX 服务器的资源的。

1. 创建 Cluster 中的资源池

创建 Cluster 的资源池或是修改其中的参数，和单机的做法完全一样，只要在 Cluster 中右击

选择 New Resource Pool 即可。而在资源池右击选择 Settings 即可修改其中的参数。

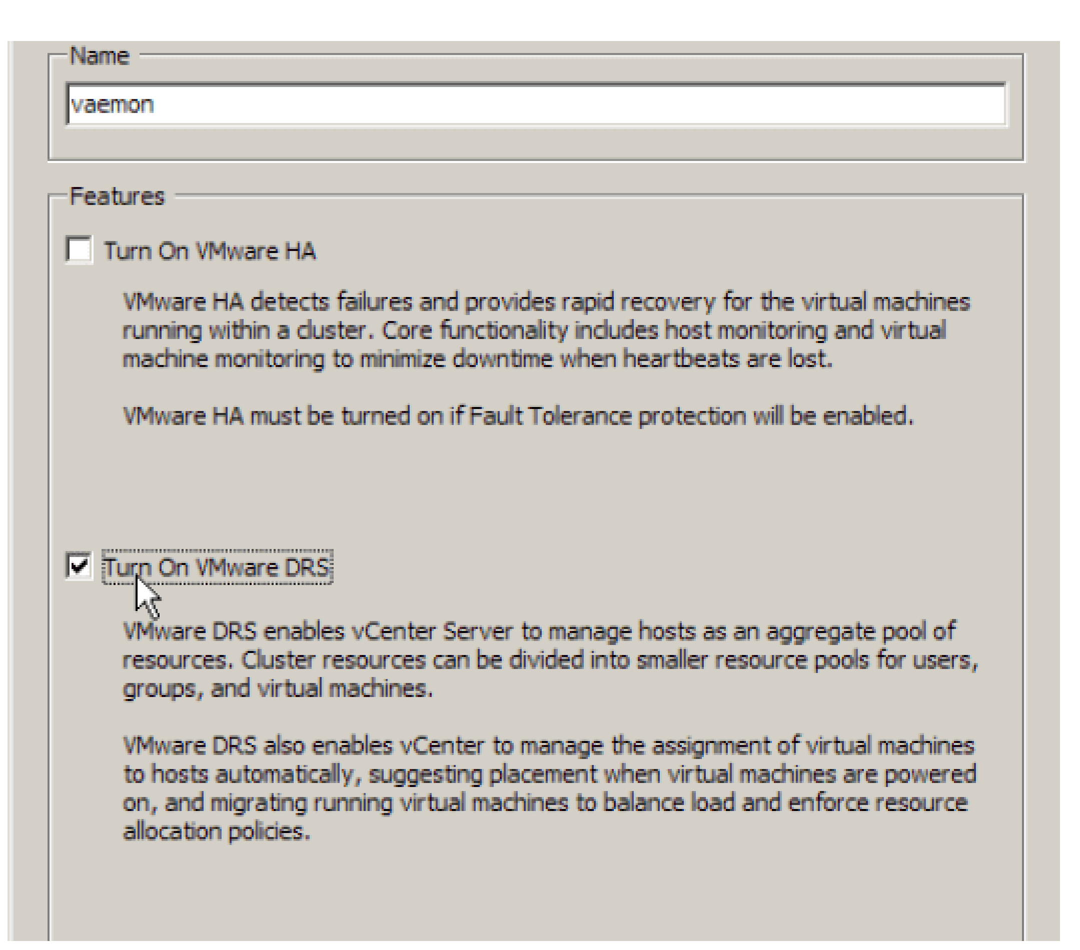
2. Cluster 中的资源池操作

在 Cluster 中的资源池操作和单机 ESX 中的资源池完全一样，但你会发现不管是 CPU 或是内存，都是资源池中所有 ESX 物理服务器的总和。大家要有一个概念，就是资源池是给多台 VM 使用的，因此资源池中的制约或是保留，都是针对在这个资源池中的多台 VM 来规范。

16.2 vSphere 最重要的功能：分布式资源调度器 DRS

DRS（Distributed Resource Scheduler）是 vCenter Server 中最重要的功能之一，主要的目的就是在集群中的多台 ESX 主机之间来分派不同的虚拟机负担。DRS 最主要的两个功能，第一是虚拟机激活时，是由集群中的哪一个 ESX 主机来承载这个虚拟机。第二个功能是评估集群上的负载，经过评估之后，会对集群中的虚拟机和 ESX 主机之间进行建议，甚至是直接将虚拟机移到最适合的 ESX 主机上，这个移动当然是使用 VMotion。当然 DRS 也允许系统管理员手动来处理这些虚拟机和 ESX 主机之间的负担和移动。

要激活 DRS 相当简单，只要在想要激活的集群上右击，选择 Edit Settings，并且将 DRS 功能选中即可。但在选中之后会有相当多的选项，我们先来理解这些选项，再来使用 DRS 的功能。接下来我们就先理解一下 DRS 的运作原理。

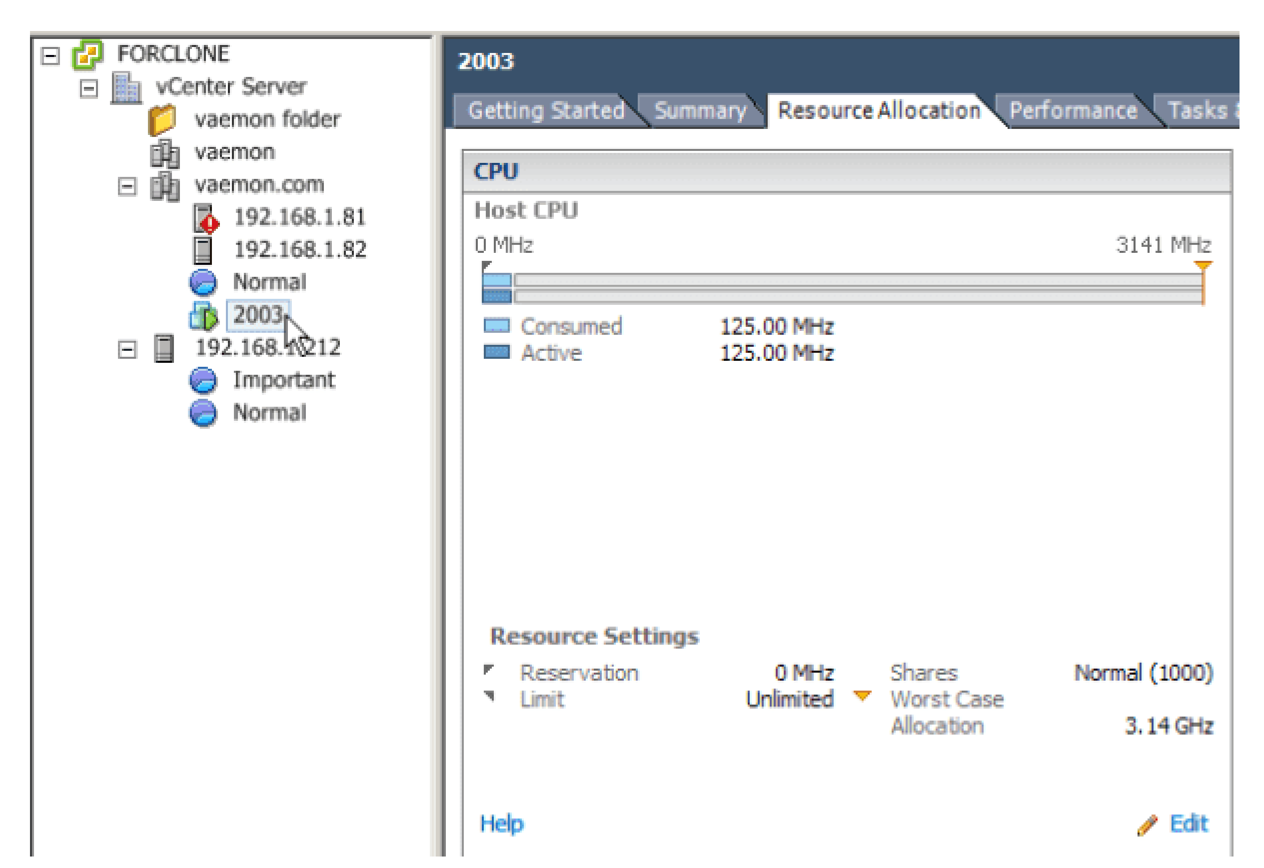


▲ 可以将任何集群给定成具有 DRS 功能

16.2.1 DRS 的运作原理

这里要提出一个十分重要的观念，就是在 DRS 激活的集群中，这个集群中的所有 ESX 主机，将不再有自身专属的虚拟机了。举例来说，原来在 192.168.1.81 所创建的 2003 虚拟机在 192.168.1.81 加入了 DRS 集群之后，这个虚拟机 2003 将无法再归属于任何 ESX 主机，因为在 DRS 的观念中，虚拟机是可以在集群中的 ESX 主机之间找到最适合的落脚处，当虚拟机激活时，会有不同的选项来查找 ESX 主机来放置这个虚拟机，因此我们会在 DRS 激活的集群中，不再看到虚拟机位于 ESX 主机下，而是和 ESX 主机的并行位置。当然在具体的属性上，你的虚拟机在哪一

台 ESX 主机上创建的，其实文件仍然是保留在该 ESX 主机上，只是已经不再有任何绑定的观念了。



▲ 在 DRS 的集群中，虚拟机已经不再专属于任何 ESX 主机了

在创建集群时，我们可以看到选择 DRS 的画面。当选择了 DRS 画面之后，就会有一些参数配置。这些就是激活 DRS 的开关。当某台虚拟机的负担大于一个程度之后，DRS 是否激活要视这些开关的配置、自动化的程度。这些开关主要有三个，分别为手动（Manual）、半自动（Partially automated）及全自动（Fully automated），我们就分别来看看。

▲ 激活 DRS 后就会有许多选项必须配置，主要是 DRS 的方式

1. 手动（Manual）的 DRS

当这个集群的 DRS 自动化参数配置成手动时，每一次集群中的 ESX 主机上的虚拟机激活时，系统就会提示你要在哪一台可用的 ESX 上激活这台虚拟机。在激活时，系统会给出可用的主机，并且以优先序推荐（Priority Recommendations）的等级来排序，最适合的 ESX 主机会在最上面。

当 DRS 嗅探到 ESX 主机之间的不平衡时，也会自动提出建议让用户来移动虚拟机。但 DRS 的激活时机一定是在 5 分钟之内发现这台虚拟机的效能变得很低才会有动作，并且也会给出可以使用的主机要求你来移动这个主机。

2. 半自动（Partially animated）的 DRS

选择半自动的 DRS 时，系统会自动选择虚拟机最佳的主机，不会让用户来选择主机，但是在真正进行移转动作之前，还是会请用户回车。

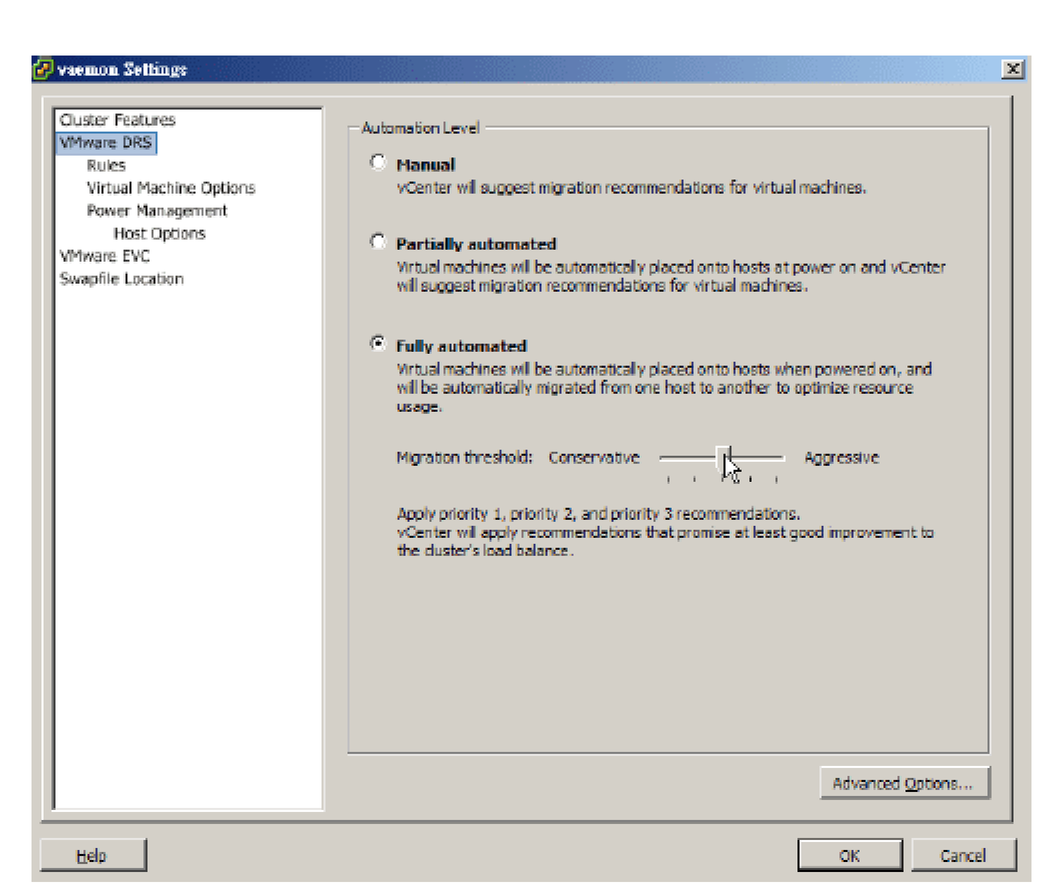
3. 全自动（Fully animated）的 DRS

全自动的 DRS 会在虚拟机激活时自身找到最佳的 ESX 主机，并且根据配置值激活 VMotion 来移转主机。全自动 DRS 的配置值共有五个，主要是以拉动条来选择。下面就是五种选择：

▶ 全自动 DRS 的五个选项

- 最保守（Conservative）：当主机是优先序 1 的推荐时才进行移转，就是所有的规则都匹配时才会进行移转，这个等级通常是要对主机进行维修了，才会要求转移，几乎不移动主机。

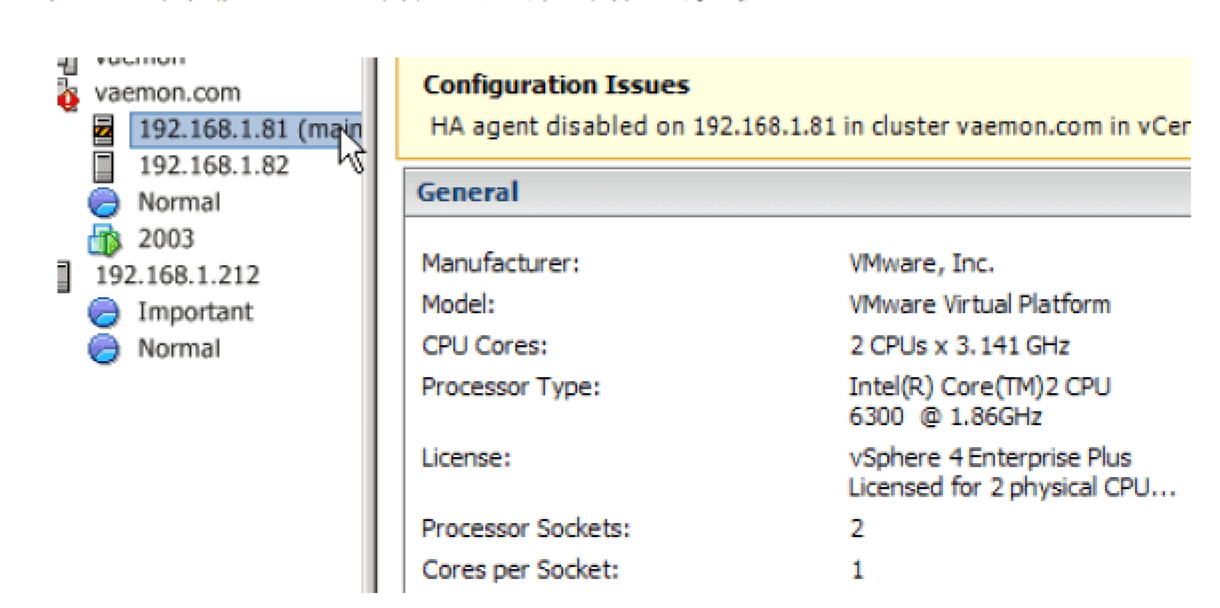
- 相对保守：当系统是 1 或 2 等级的优先序推荐时才会进行转移。其他推荐以下的会要求用户回车。当系统真的会有很显著的效能提高时才会转移。
- 中等：当系统是 1、2 或 3 等级的优先序推荐时才会进行转移。至少要有不错的效能进步时才会转移。
- 积极：当系统是 1、2、3 或 4 等级的优先序推荐时才会进行转移，有不差的效能进步时才会进行移转。
- Aggressive：接受所有建议，只要有一点点效能进步就会移转，不经过用户回车。



▲ 全自动的 DRS 也有不同的嗅探等级

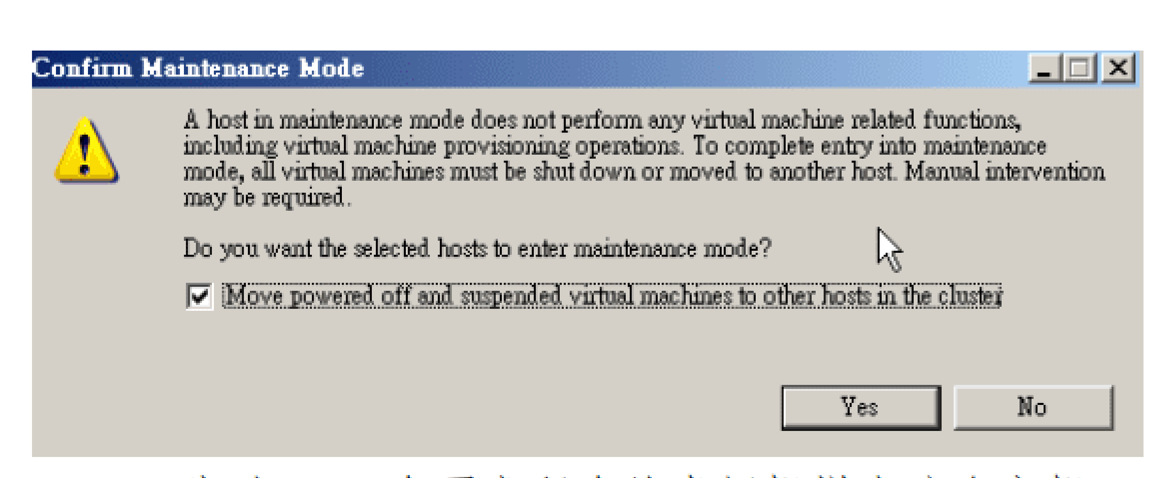
4. 完全不建议的特殊情况：维护模式

在特殊的情况下，我们需要将一台 ESX 主机进行维修（如热插拔内存或 CPU）时，我们可以在 vCenter Server 中在该主机上右击，选择 Enter Maintenance Mode。当一台 ESX 主机在 DRS 集群中进入维护模式之后，这台机器就不可以以任何方式将虚拟机移入了。因为这台主机都在维护了，怎么可能还可以再让别的主机移入呢？因此这台主机将会被排除在外，但其上的虚拟机还是可以继续进行，只是无法再移入、新建或激活虚拟机。



▲ 主机进入维护模式时，意味着这台主机将不能运行任何虚拟机

另外这台主机如果被配置成维护模式，其上的虚拟机会立即被建议五颗星的转移，换句话说，这台主机上的所有虚拟机都必须转移出去，因为这台主机正要被维修，将无法再使用了。



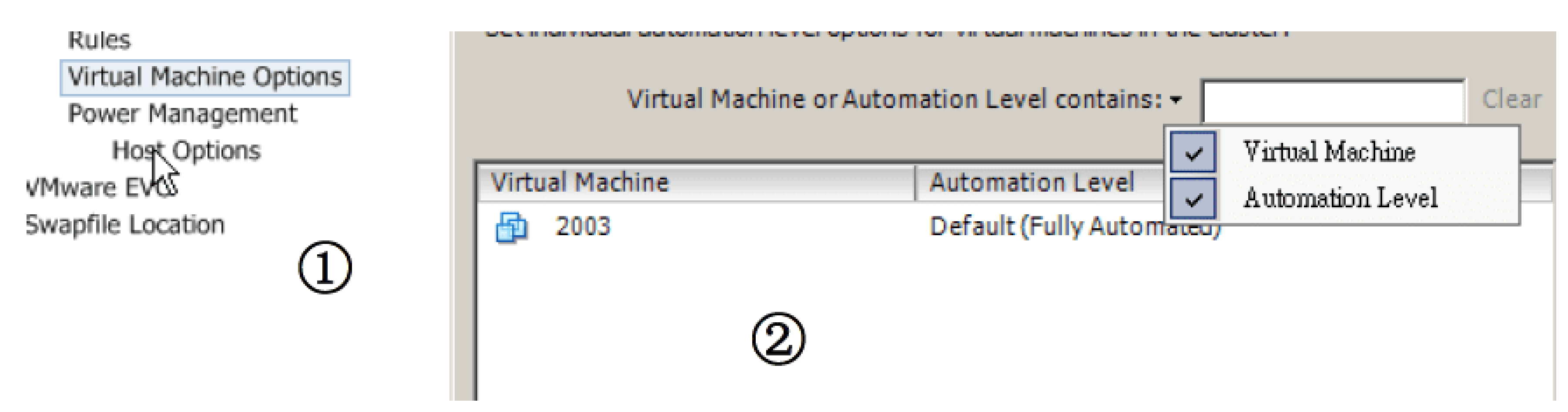
▲ 此时 DRS 会要求所有的虚拟机撤出这台主机

5. 针对单个虚拟机的配置

当虚拟机被加入 DRS 的集群之后，除了默认的 DRS 激活参数之外，我们也可以针对每一台虚拟机来配置 DRS 的激活参数，下面就是详细的配置手段。

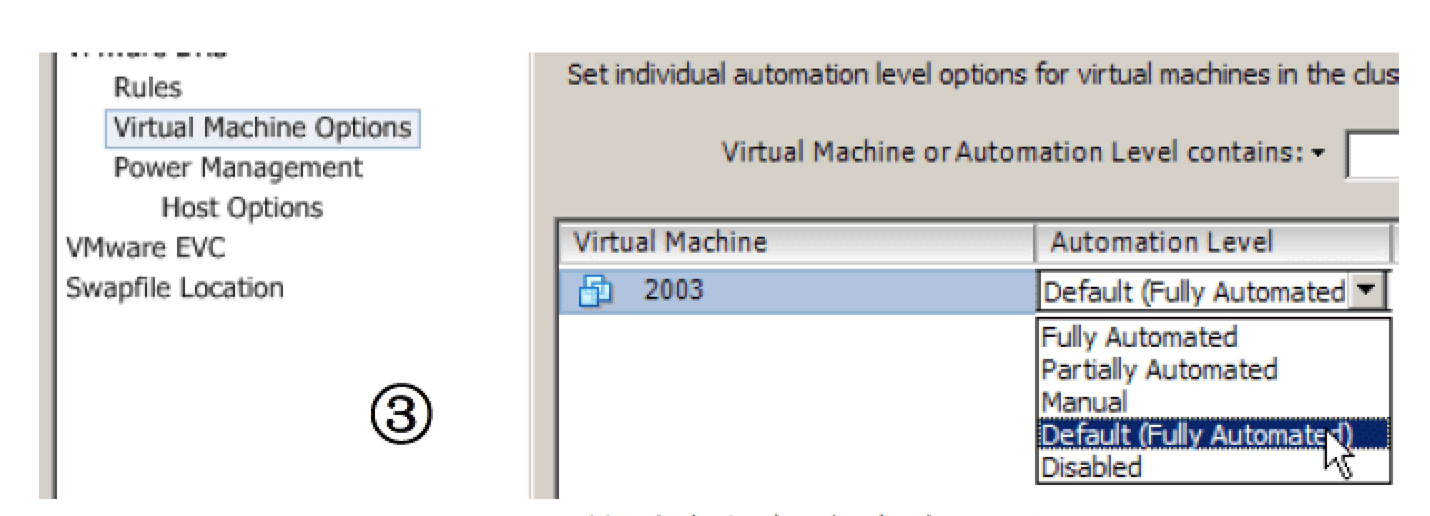
▶ 配置 DRS 集群中单个虚拟机的激活参数

1. 在已经拥有虚拟机的 DRS 集群上右击选择 Edit settings。
2. 选择 Virtual Machine Options。在这个 DRS 集群中所有的虚拟机都会给出来。



▲ 选择虚拟机的配置

3. 此时我们进入任何一台虚拟机，并且在 Automation Level 的地方有一个下拉式菜单来决定单个虚拟机的激活 DRS 配置。



▲ 可以针对虚拟机本身来配置

这里一共有五个配置，除了前面提到的三个之外，Default 值就是整体 DRS 的原始配置，而 Disable 就是停用 DRS。

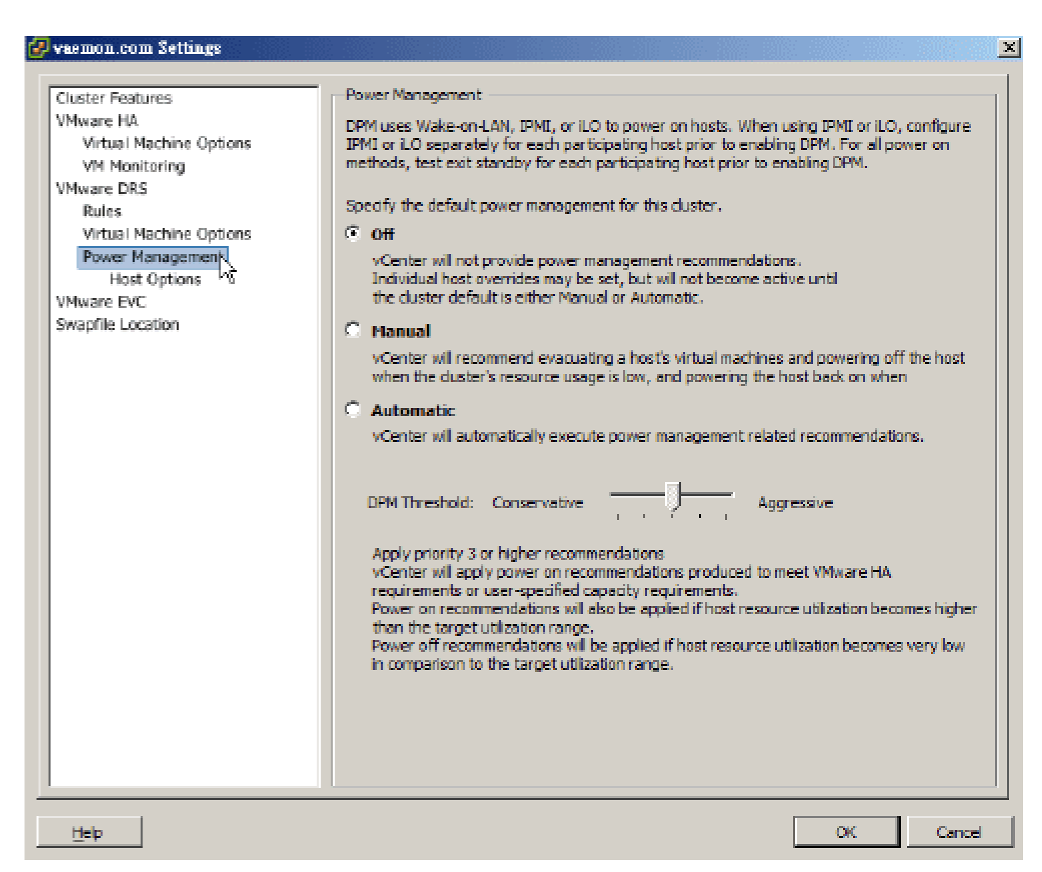
6. vSphere 的新功能：省电模式

除了前面提到的 DRS 激活模式、规则之外，DRS 的激活时机在新版的 vSphere 中添加了一项省电模式。当整个 DRS 集群的系统负担降低到一个程度之后，DRS 会将所有的虚拟机移到某一台 ESX 主机，并且让其他在 DRS 集群中的 ESX 主机进入省电模式。当我们进入 DRS 的集群时可以看到 Power Management 的选项，下面就是三种模式的说明。

▶ DRS 集群的省电模式

- 关闭（OFF）：vCenter Server 将不会建议电源管理，就算是独立的 ESX 上面有配置电源管理，但 DRS 并不会激活电源管理。换句话说，就算是当前所有的 ESX 主机都没有任何负担，系统仍然会全速运行。
- 手动（Manual）：当系统的负担较低时，DRS 会提出建议，并且让用户手动激活 DRS 功能，将虚拟机集中在一台 ESX 主机上，并且将没有负担的主机进入休眠模式。
- 自动（Automatic）：系统将会使用 DRS 及 VMotion 功能将所有的虚拟机集中在一台 ESX 主机上，并且让空的 ESX 主机进入休眠模式。

我们也可以独立配置每一台 ESX 主机自身的电源管理模式，但前提是 DRS 的电源管理模式必须不能是关闭（OFF）的。



▲　新版的 vSphere 可以让主机进入省电模式

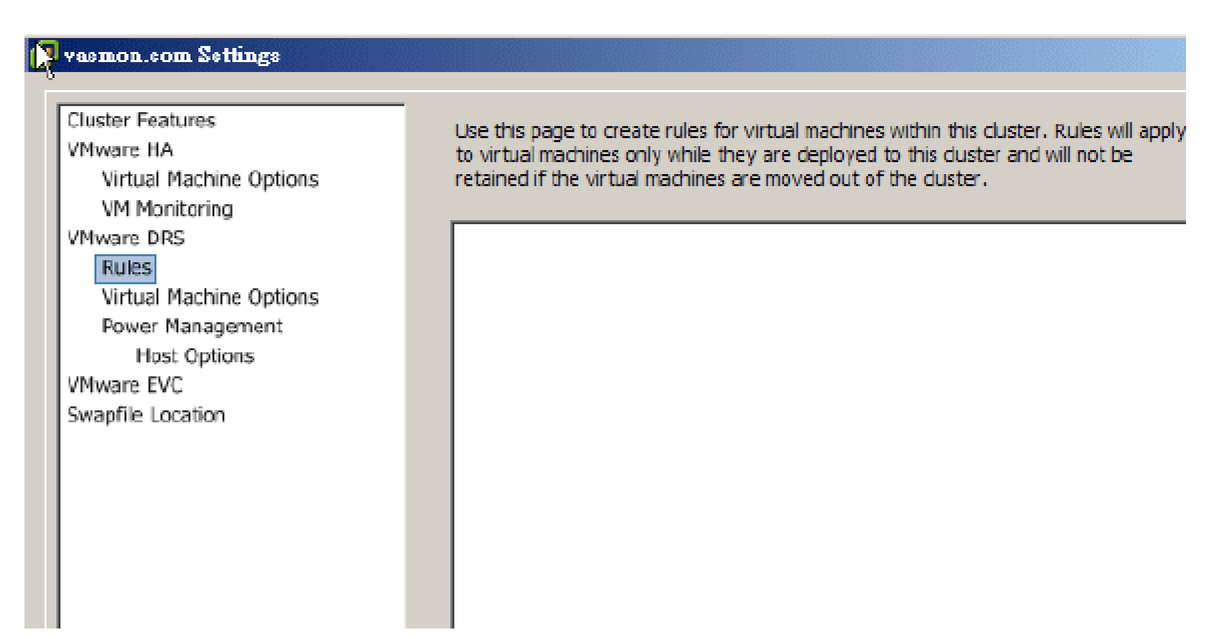
16.2.2　DRS 的规则配置（Rules）

企业中 ESX 主机上并发运行的虚拟机种类和数量，有时不是光靠效能来判断的，必须加入业务和具体需要的考虑。举例来说，在 Linux 下架设 Web 2.0 的网站，常常使用所谓的 LAMP 架构。LAMP 就是 Linux、Apache、MySQL 和 PHP。

在虚拟机世界中，LAMP 架构可以存在于两台虚拟机上，在一台 Linux 虚拟机上安装 Apache 和 PHP，而 MySQL 数据库则是安装在另一台 Linux 虚拟机上。但为了效能和方便性，我们会希望这两台 Linux 的虚拟机都安装在同一台物理 ESX 上，而不要被 DRS 的效能建议影响，将两台虚拟机在 DRS 的安排之下放到不同的 ESX 主机上。

DRS 当然也考虑到业务上的情况，因此也可以配置哪些虚拟机必须是在一起，或是哪些虚拟机是不能在一起的（如一台 MS-SQL 和一台 Exchange，因为两者的负担都较重）。如此一来，除了以效能为考虑因素之外，也加入了业务的考虑因素，让整个 DRS 变得更加弹性化。

DRS 的规则分为两类，第一类就是“将虚拟机放在一起”（Keep Virtual Machines Together）。第二类就是“不将虚拟机放在一起”（Separate Virtual Machines）。



▲　DRS 的规则可以让虚拟机绑定或是互斥

1. 将虚拟机放在一起

我们可以在 DRS 的规则中配置哪些虚拟机必须放在一起，这个选择十分简单，只要创建一条规则，并且选择哪些虚拟机要放在一起即可。要注意的是，这些规则是可以创建后不运行，只在 DRS 的配置中，不将这个规则打勾即可。如果我们配置虚拟机 A 和虚拟机 B 一定要在一起，当 DRS 在运作时，就会将这 A、B 两个虚拟机一起打包转移。

2. 不将虚拟机放在一起

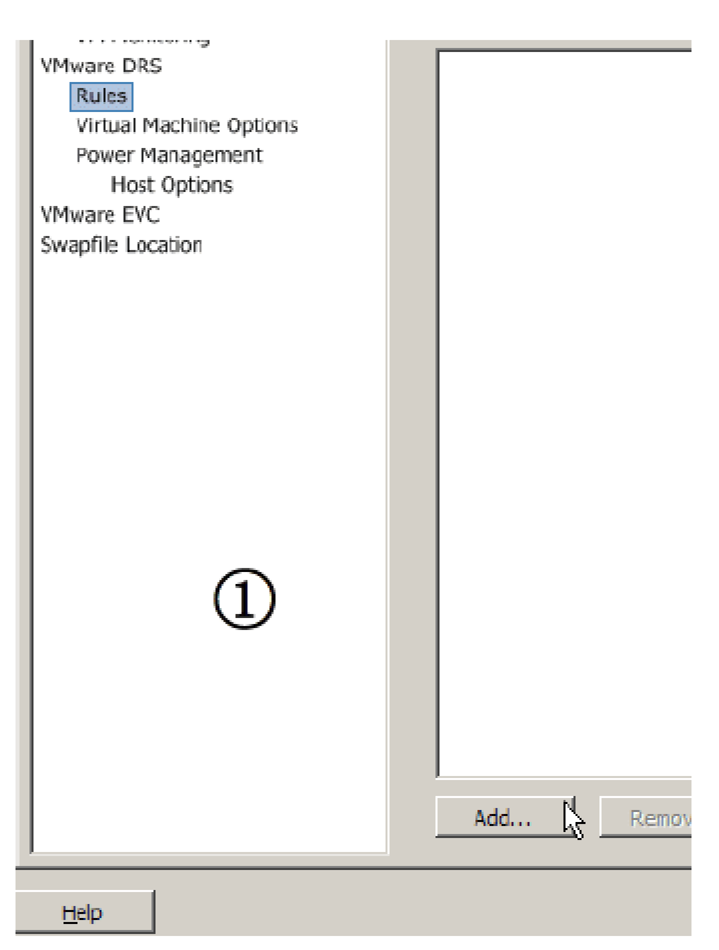
创建这一类的规则，可以将不要放在一起的虚拟机选中，如果我们将虚拟机 C 和 D 配置不能在一起，当 DRS 运作时，如果发现虚拟机 C 的目的 192.168.1.81 主机上已经有 D 了，就不会将 192.168.1.81 这台主机列为虚拟机 C 的候选目的主机。

3. 创建 DRS 规则实例

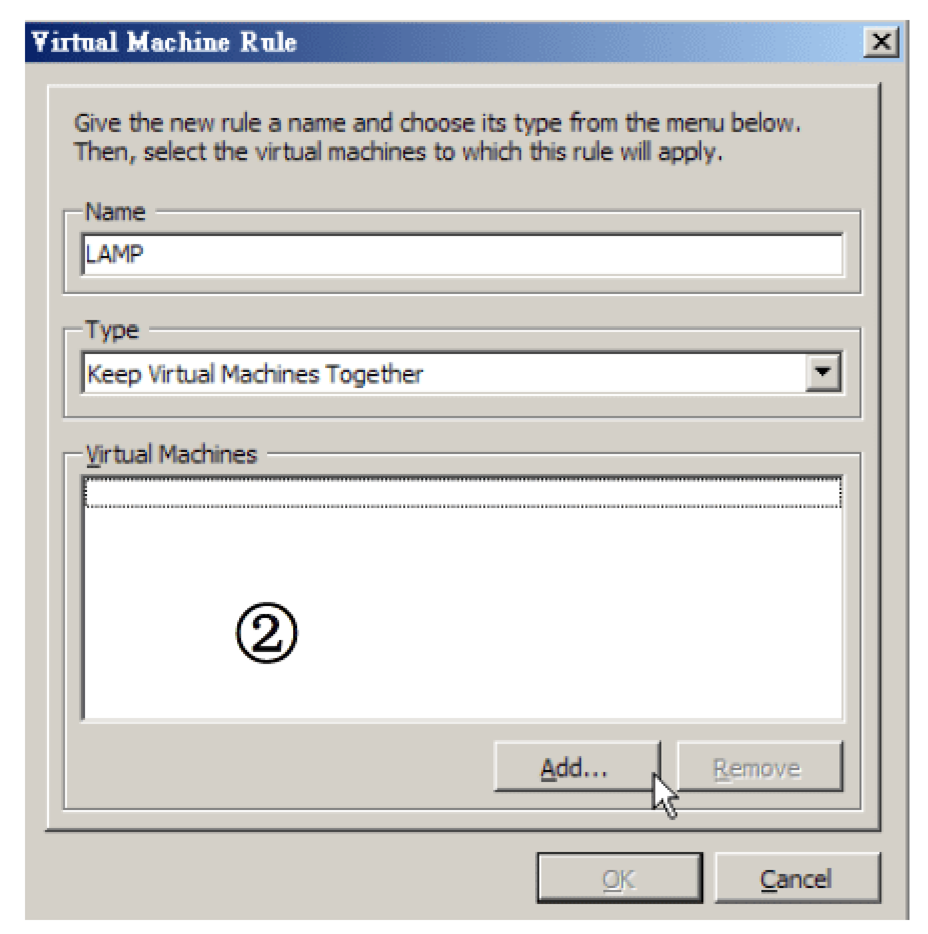
创建 DRS 的规则再简单不过了，只要选择你要让虚拟机在一起或不能在一起，再将选中的虚拟机加入即可。下面就是一个简单的例子。

▶ 创建 DRS 规则

1. 进入 DRS 集群的配置部分，并且选择 Rules。单击 Add 按钮。
2. 首先我们键入这个规则的名称（如 LAMP），并且选择让虚拟机在一起的选项。之后单击 Add 按钮加入虚拟机。

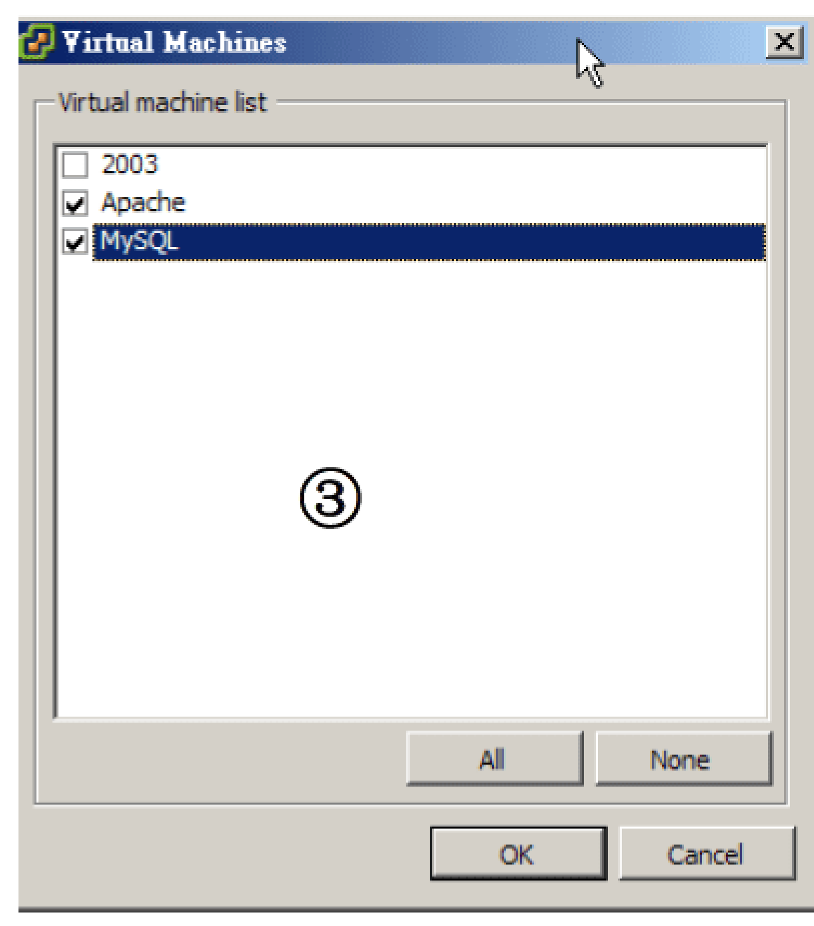


▲ 创建新的规则

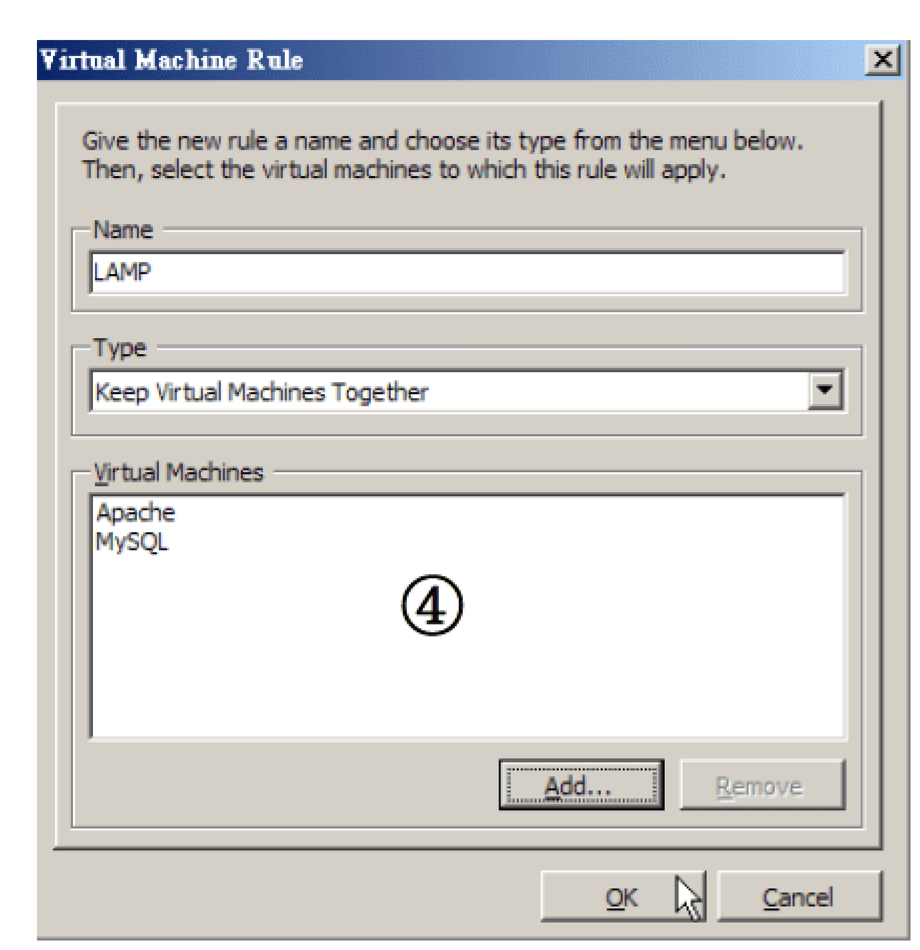


▲ 键入名称之后来加入虚拟机

3. 选中要加入这项规则的虚拟机，在选择完后单击 OK 按钮。
4. 此时我们会看到选中的虚拟机。单击 OK 按钮。

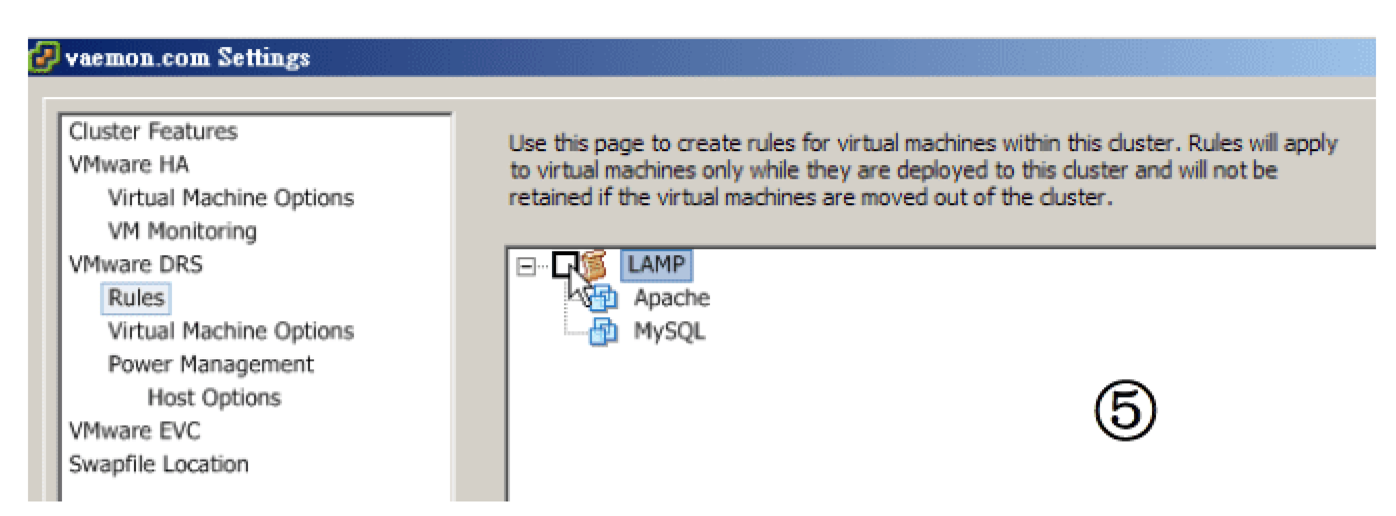


▲ 选中要加入的虚拟机



▲ 选中的虚拟机

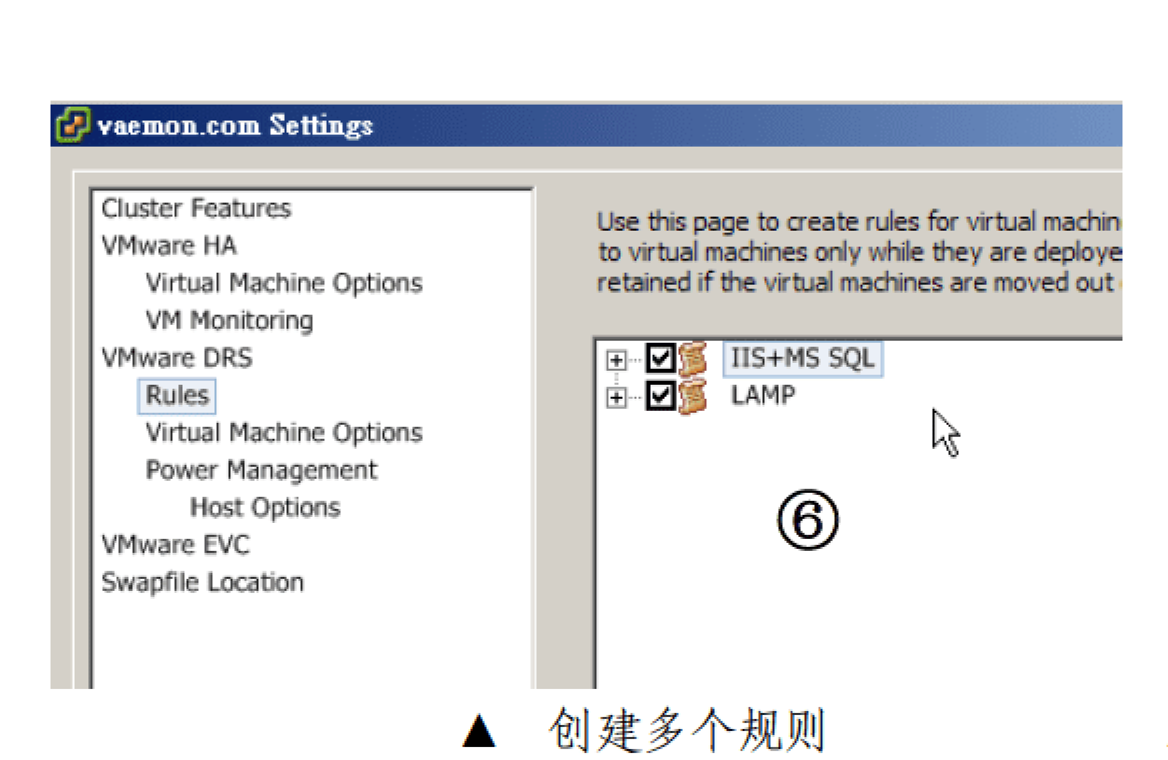
5. 在 DRS 集群的 Rules 窗口中，我们就看到了这个规则了。如果你希望这个规则暂时不被应用，也可以将其上的打勾撤销。



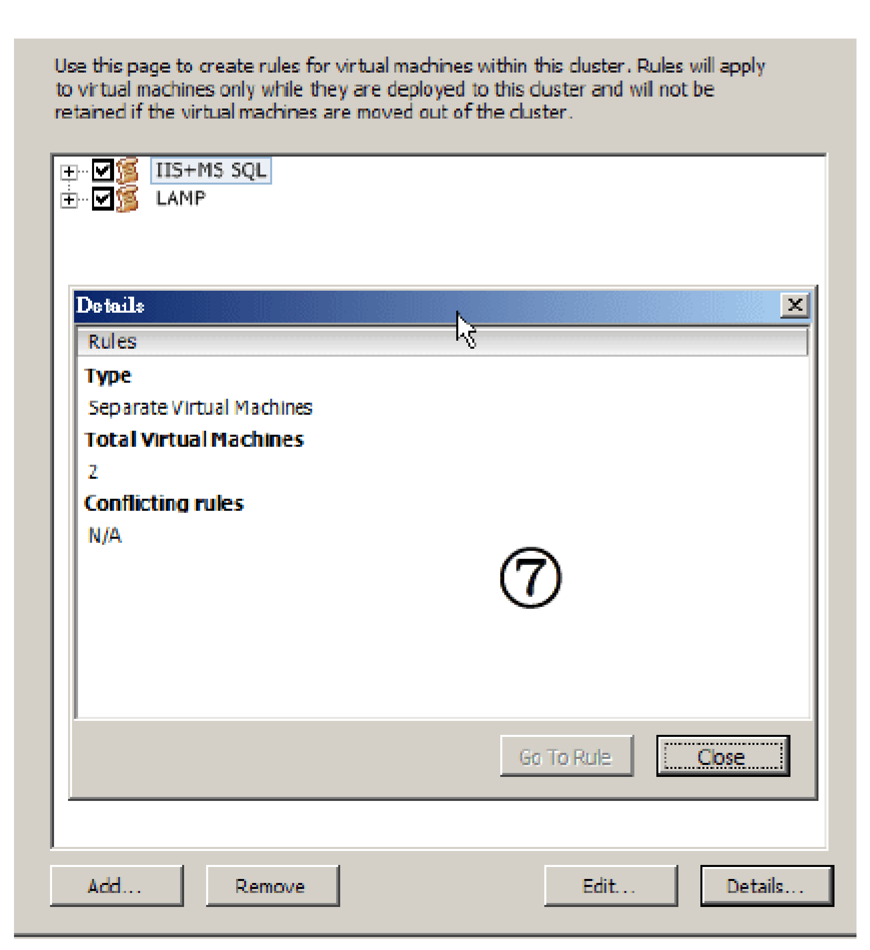
▲ 可以将整个规则暂时撤销

6. 我们也可以配置一个规则，就是 SQL 和 Redhat 两个主机不能在一起。

7. vSphere 的 DRS 规则相当智能，会遍历创建的规则是否合乎逻辑，举例来说，如果我们创建两个规则是相斥的，系统会主动告知。在创建规则之后，可以单击 Detail 按钮来检视是否和其他规则相冲。



▲ 创建多个规则



▲ 如果有两个规则相冲，DRS 也会嗅探出来

16.2.3 在 vCenter Server 中实践 DRS

DRS 的运作前提，是必须将 ESX 主机放入一个激活 DRS 功能的集群中，而本小节开宗明义地说过，如果你的 ESX 主机上已经有虚拟机了，那么在将这台 ESX 主机加入 DRS 的集群之后，这些虚拟机将不再是某个 ESX 专属的了，因为这些虚拟机已经会“流来流去”。我们在这一小节就来看看 DRS 的实践。

1. 创建具有 DRS 功能的集群

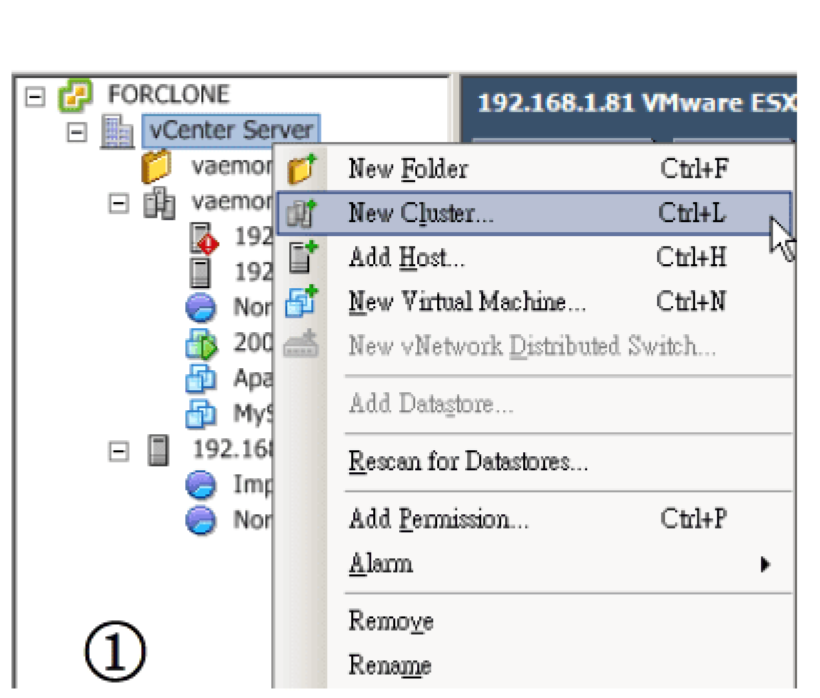
创建具有 DRS 的集群和创建一般的集群一样，只要在创建时激活 DRS 选项，并且配置前面所提到的参数即可。下面就是详细的步骤。

▶ 创建具有 DRS 功能的集群

1. 在 vCenter Server 中的 Datacetner vSphere 上右击，选择 New Cluster 选项。

2. 在如下图所示的画面中，键入这个集群的名称，如 vaemon。别忘了将 VMware DRS

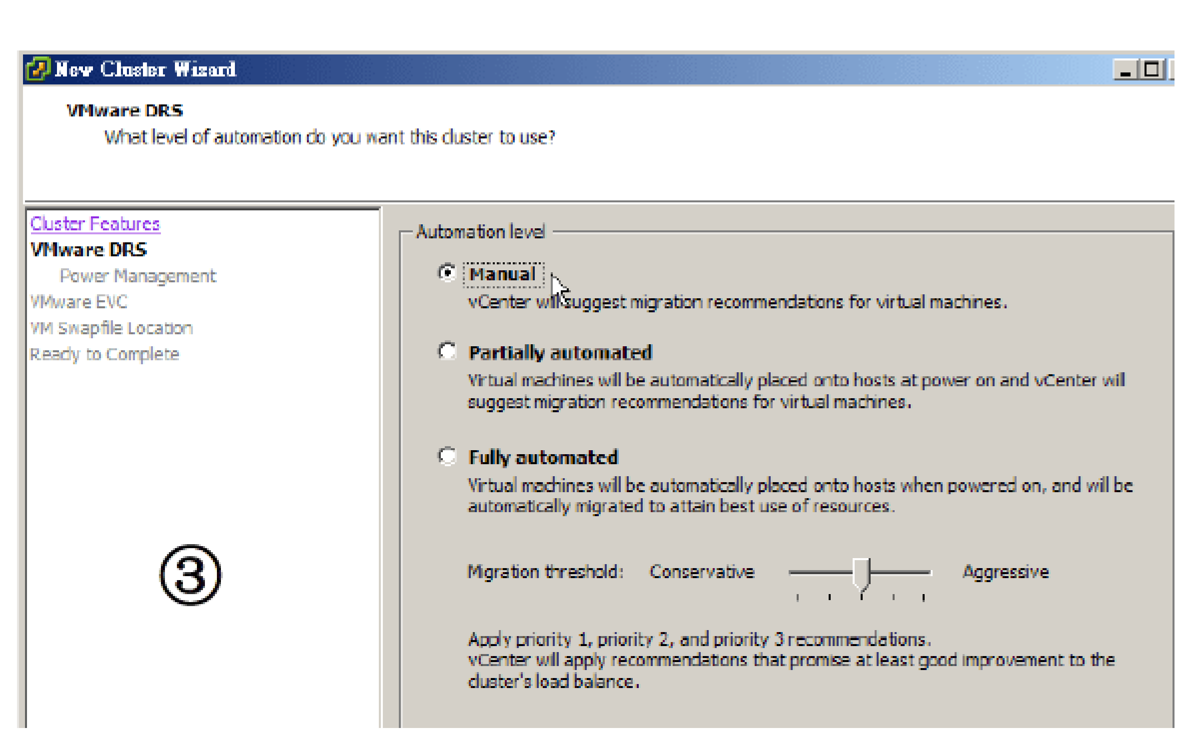
的选项打勾。打勾之后，在左边的窗格会有很多 DRS 的参数弹出。



▲ 选择新建集群　　▲ 别忘了选中 DRS 选项，左边也会有参数弹出

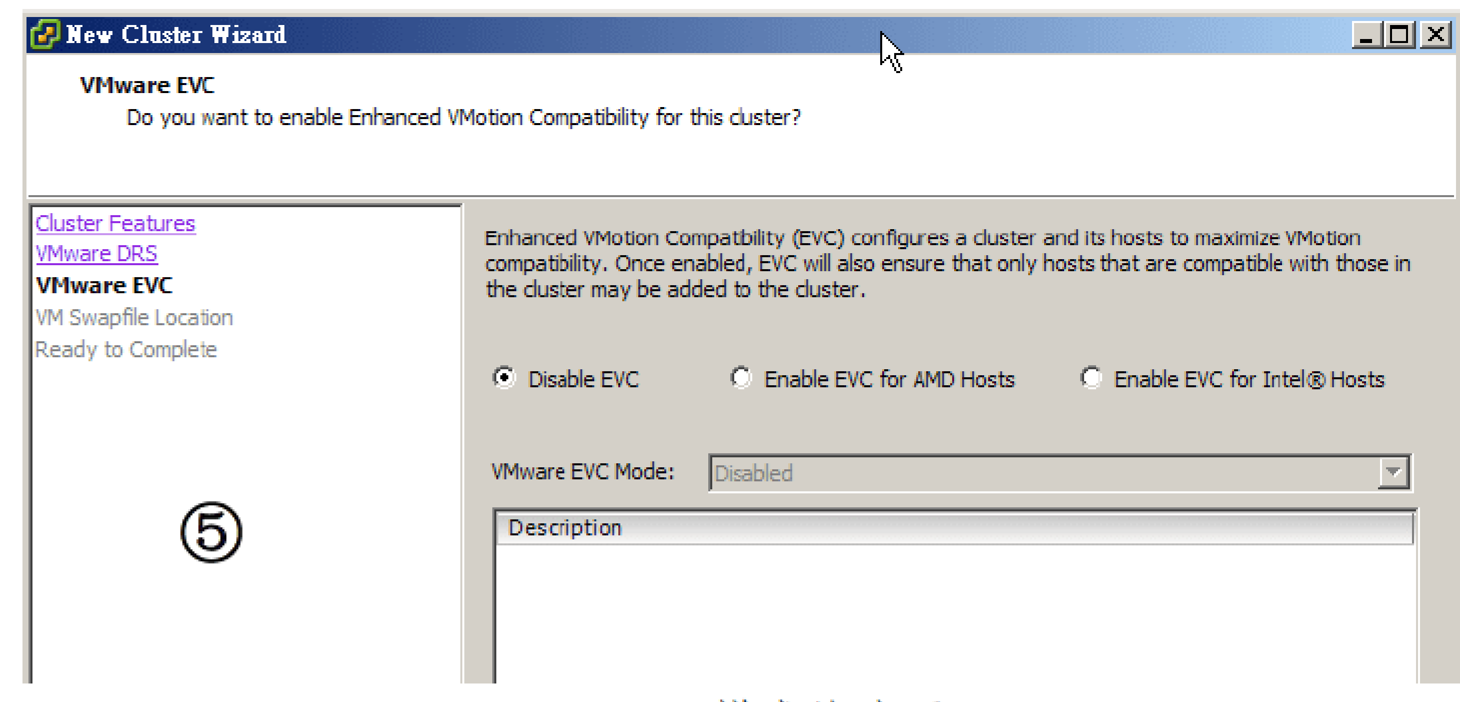
3. 接下来就是选择这个 DRS 集群的转移规则，我们先设成手动。

4. 接下来省电模式，选择默认值即可。



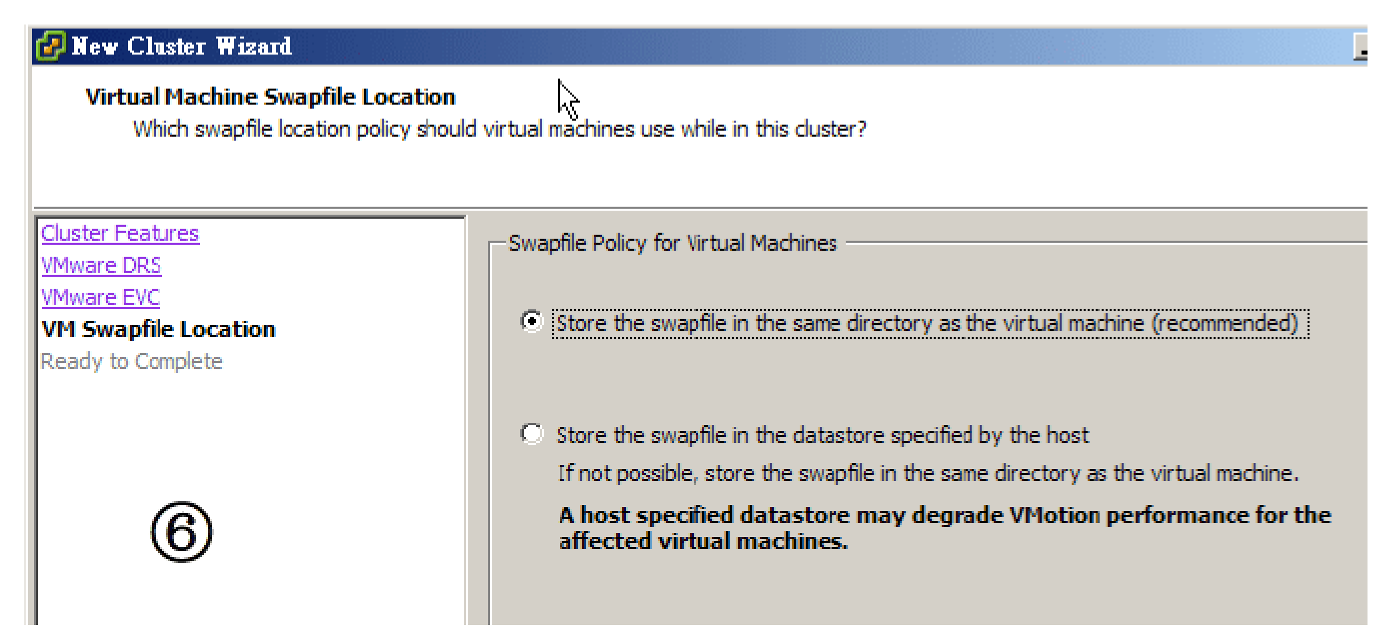
▲ 配置转移规则　　▲ 配置省电模式

5. 接下来确定 VMotion 的规则。



▲ EVC 模式的选项

6. 接下来是置换文件的位置。最后会弹出总结画面，单击 Finish 按钮后就落实了。



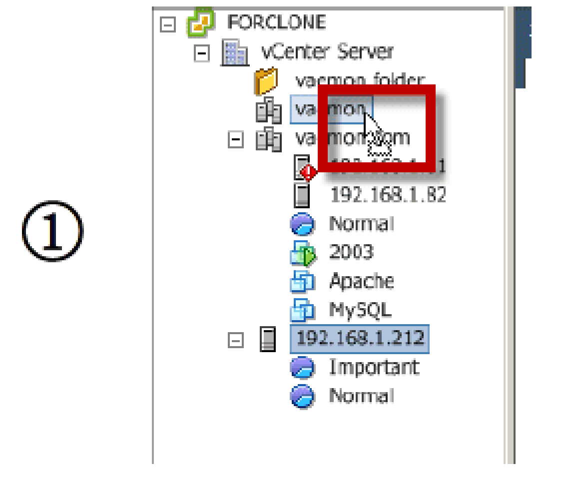
▲ 置换文件的位置

2. 将主机加入 DRS 的集群中

ESX 主机是 DRS 集群的基本单位，因此我们必须将主机加入集群中。将 ESX 加入 DRS 集群中有多种情况，包括这台 ESX 主机上当前还没虚拟机（最简单）、这台主机上已经有虚拟机了，或是这台 ESX 主机上不但有虚拟机及资源池。我们就来看看不同的情况是如何加入主机的。

▶ 将 ESX 主机加入 DRS 集群

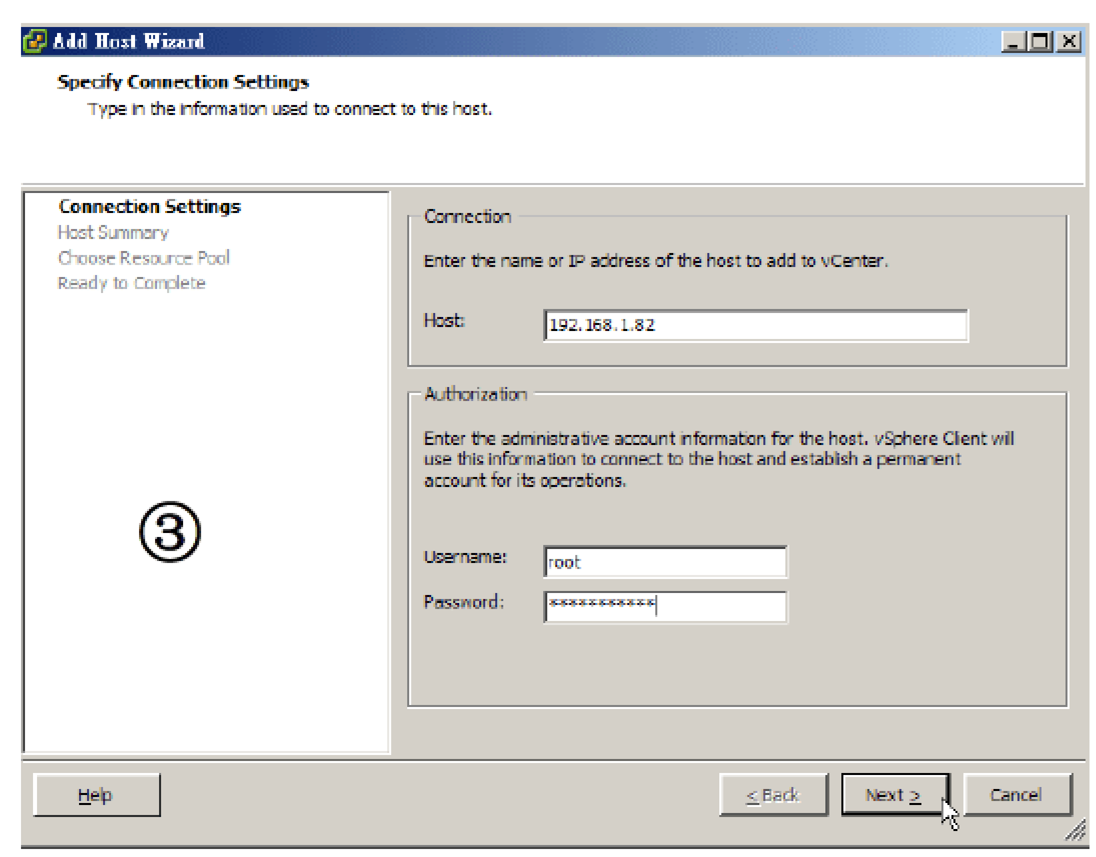
1. 在 DRS 的集群中加入 ESX 主机，前提是这个主机还没有被加入 vCenter Server。如果你想要从已经加入 vCenter Server 的主机中选择放入 DRS 集群，可以直接将主机拖拉至集群上。
2. 本例中，192.168.1.82 这台 ESX 主机上面当前没有虚拟机，首先选择 Add Host 选项。



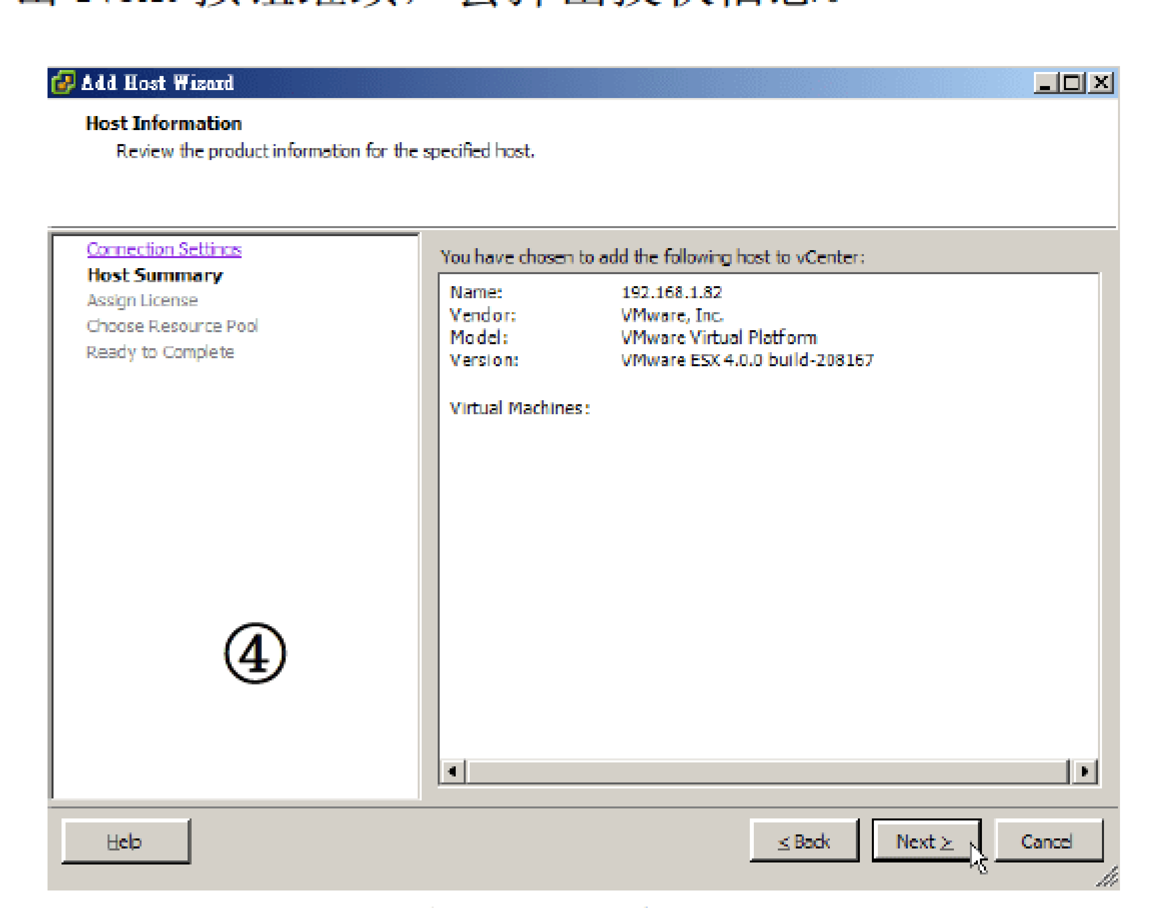
▲ 如果是已经存在于 vCenter Server 上的主机，可以用拖拉的方式

▲ 选择加入主机

3. 在如下图所示的窗口中键入主机名及账号口令等。
4. 此时会弹出这台主机的数据，直接单击 Next 按钮继续，会弹出授权信息。

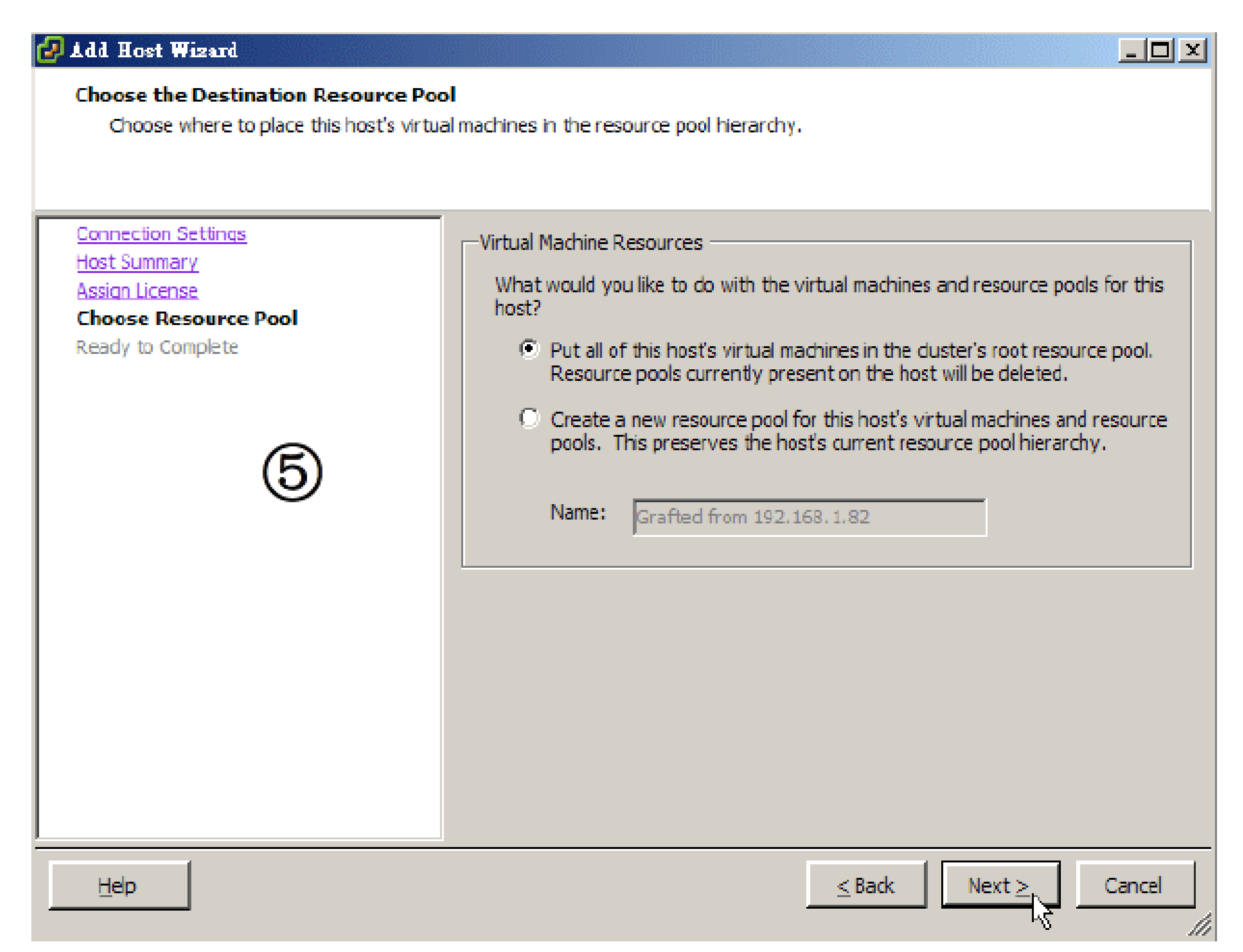


▲ 键入主机名和账号口令



▲ 会弹出这台主机的数据

5. 接下来是相当重要的配置。系统会询问你是否要将这台主机上所有的虚拟机放到集群的根目录下，或是替这台 ESX 主机，在 DRS 集群下创建一个新的资源池，但仍然保留原来资源池的配置。一般来说，我们会希望由 DRS 来分配资源，因此选择第一项。

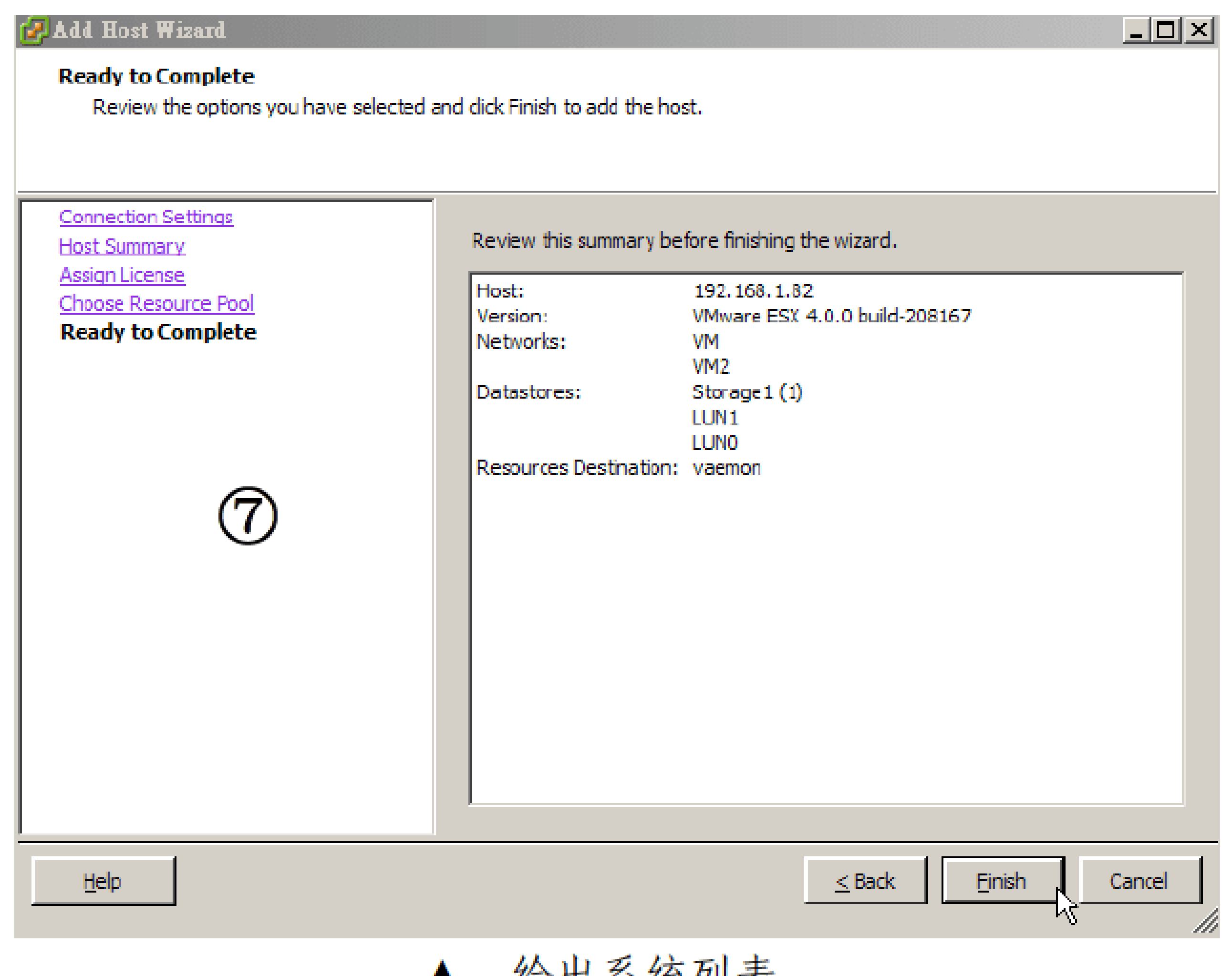


▲ 如果主机有资源池，系统会新建一个在集群之下而非主机之下

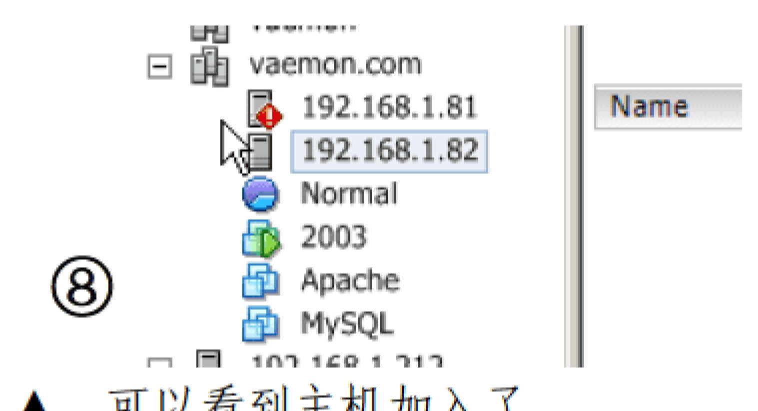
6. 如果你不希望 DRS 集群破坏你原来在 ESX 主机中创建的资源池，可以选择第二项，此时系统会在这个 DRS 集群下创建一个新的资源池，名称会是 Grafted from 192.168.1.81。

7. 接下来会弹出系统列表，单击 Finish 按钮后即可将主机加入。

8. 最后可以看到主机加入了。

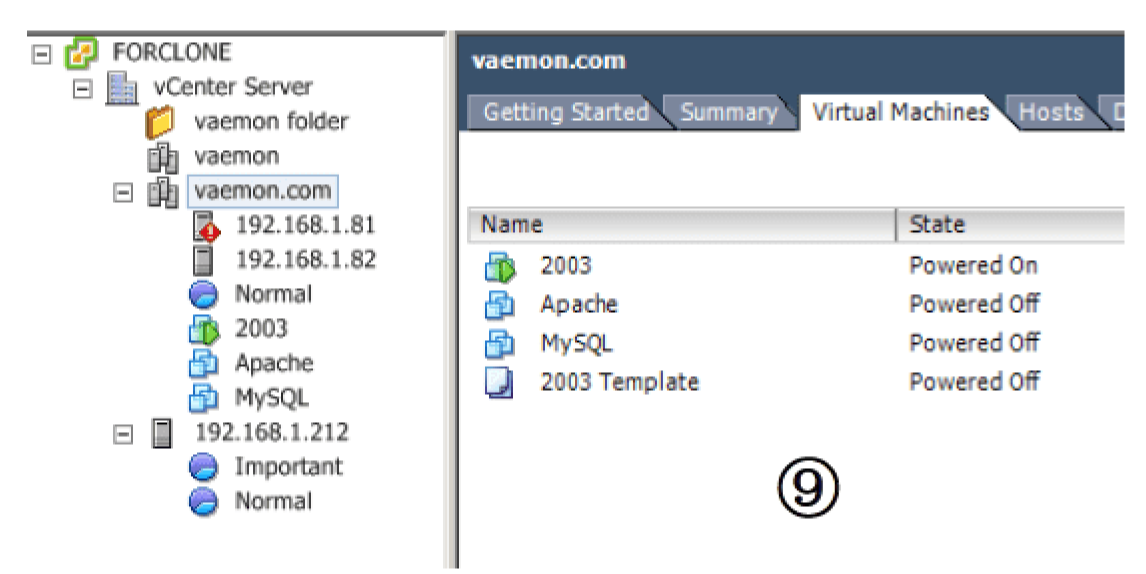


▲ 给出系统列表



▲ 可以看到主机加入了

9. 当我们将已经有虚拟机的 ESX 主机加入后，你会发现整个 192.168.1.81 上的虚拟机都被移到 DRS 集群的根目录下了。



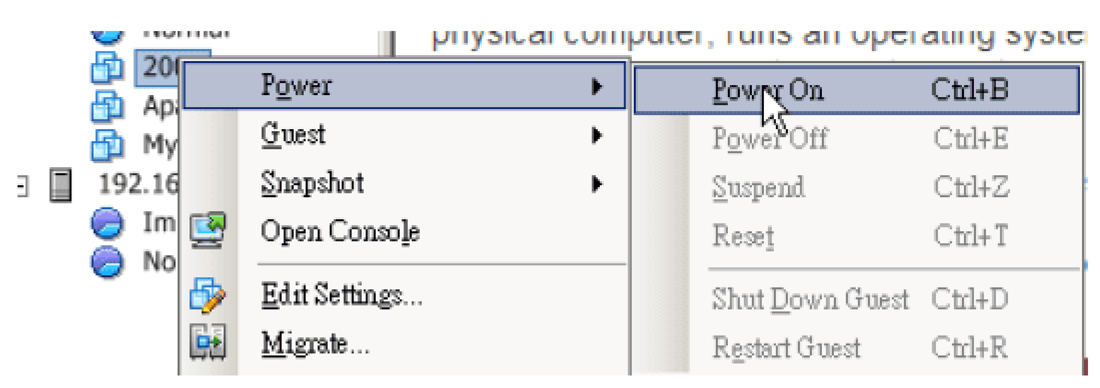
▲ 如果是有虚拟机的 ESX 主机，虚拟机都会移到集群的根目录下

3. DRS 实例：引导时的 ESX 主机选择

当我们配置手动时，任何虚拟机的引导或是负担改变，都会激活 DRS 的建议。我们就先来看看引导时的 DRS 建议的选择。

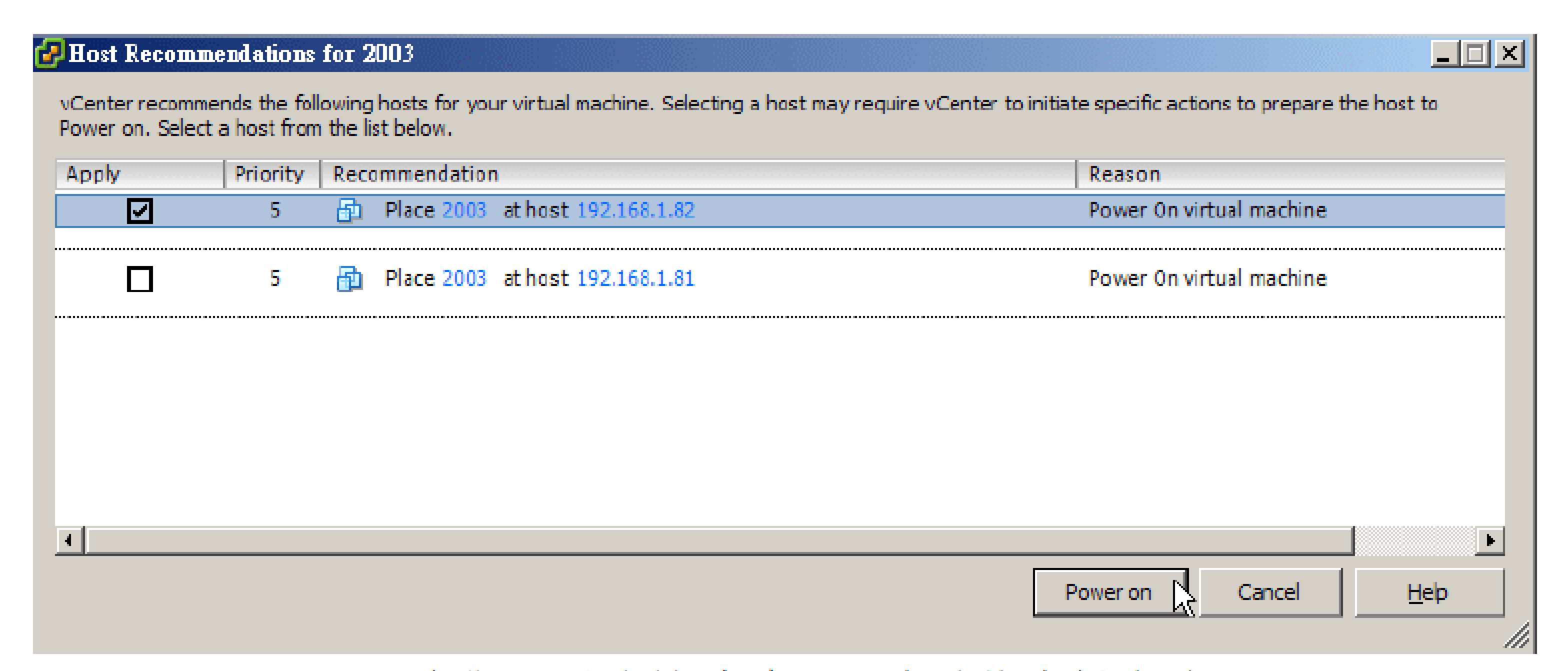
▶ 虚拟机引导时的 DRS

1. 我们进入任何一台虚拟机，并且在其上右击选择 Power On 选项。



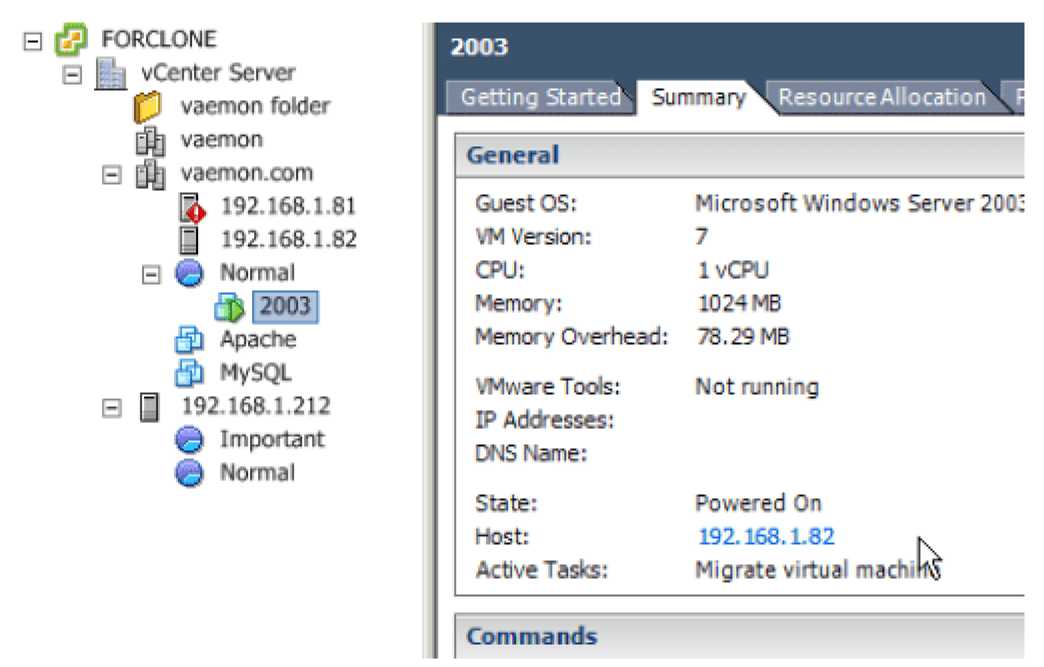
▲ 选择引导

2. 此时会弹出一个建议窗口，在没有任何负担的情况下，两台 DRS 集群中主机各获取一个新的建议，因为两台主机是完全一样，没有任何一台的负担比另一台大。当你选中了想要引导的主机之后，直接单击 Power on 按钮即可。



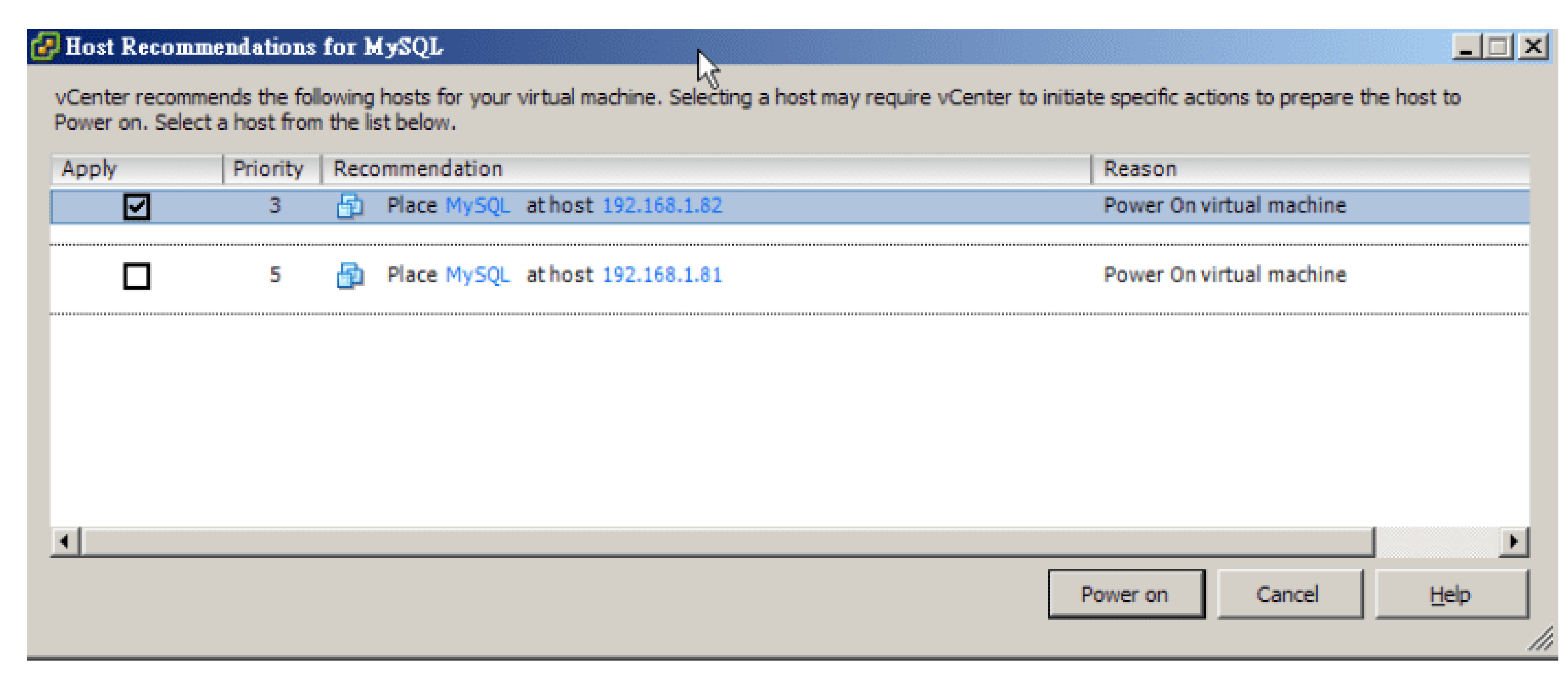
▲ 我们配置的是手动，因此系统会提出建议

3. 此时 vCenter Server 会视为你已经采取的 DRS 的建议，并且进行引导的动作。
4. 此时我们可以选择这台虚拟机的 Summary 的选项卡，可以看到当前这台虚拟机的运作 ESX 主机，并且显示当前的运作情况。



▲ 可以看到这台虚拟机已经在选择的主机上引导了

5. 如果我们再激活一台虚拟机就会看到两台 ESX 主机的等级已经改变了。



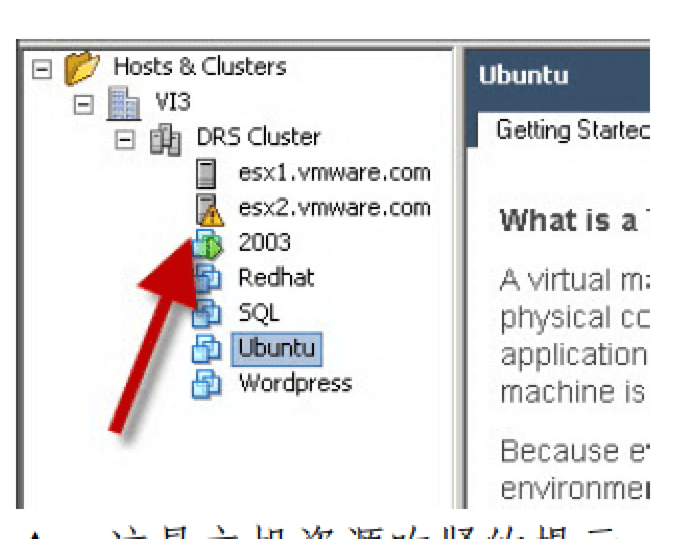
▲ 当 ESX 主机被一台虚拟机使用时，就会发生等级的改变

4. DRS 运作时的系统资源观察

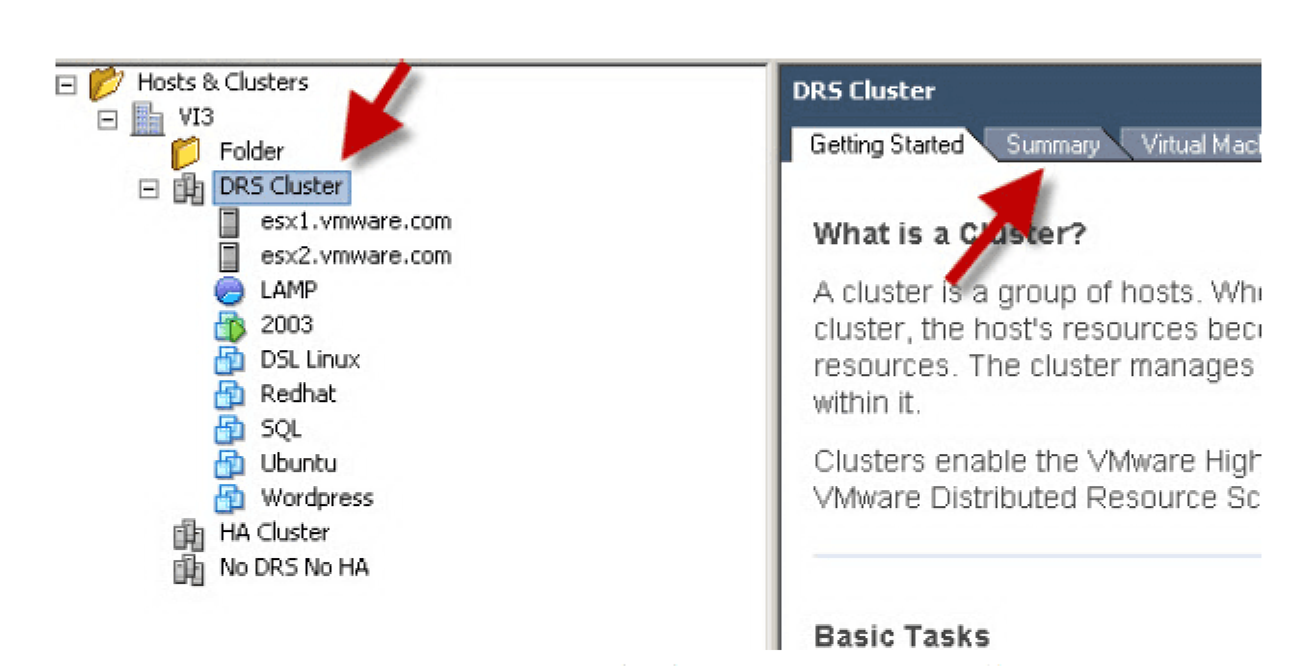
当 DRS 在集群中开始运作之后，我们可以观察在 DRS 中的资源应用情况。其中最重要的就是每一台 ESX 主机的体现。我们就来看看最重要的几个指针。

▶ DRS 运作时的主机参数

1. 当有任何 ESX 的主机资源不敷使用时，我们可以看到左边的窗格主机上会弹出一个感叹号，意味着这台主机的资源不够了。
2. 此时我们可以选择集群的名称，并且选择 Summary 选项卡来检视整个集群的系统资源使用情况。

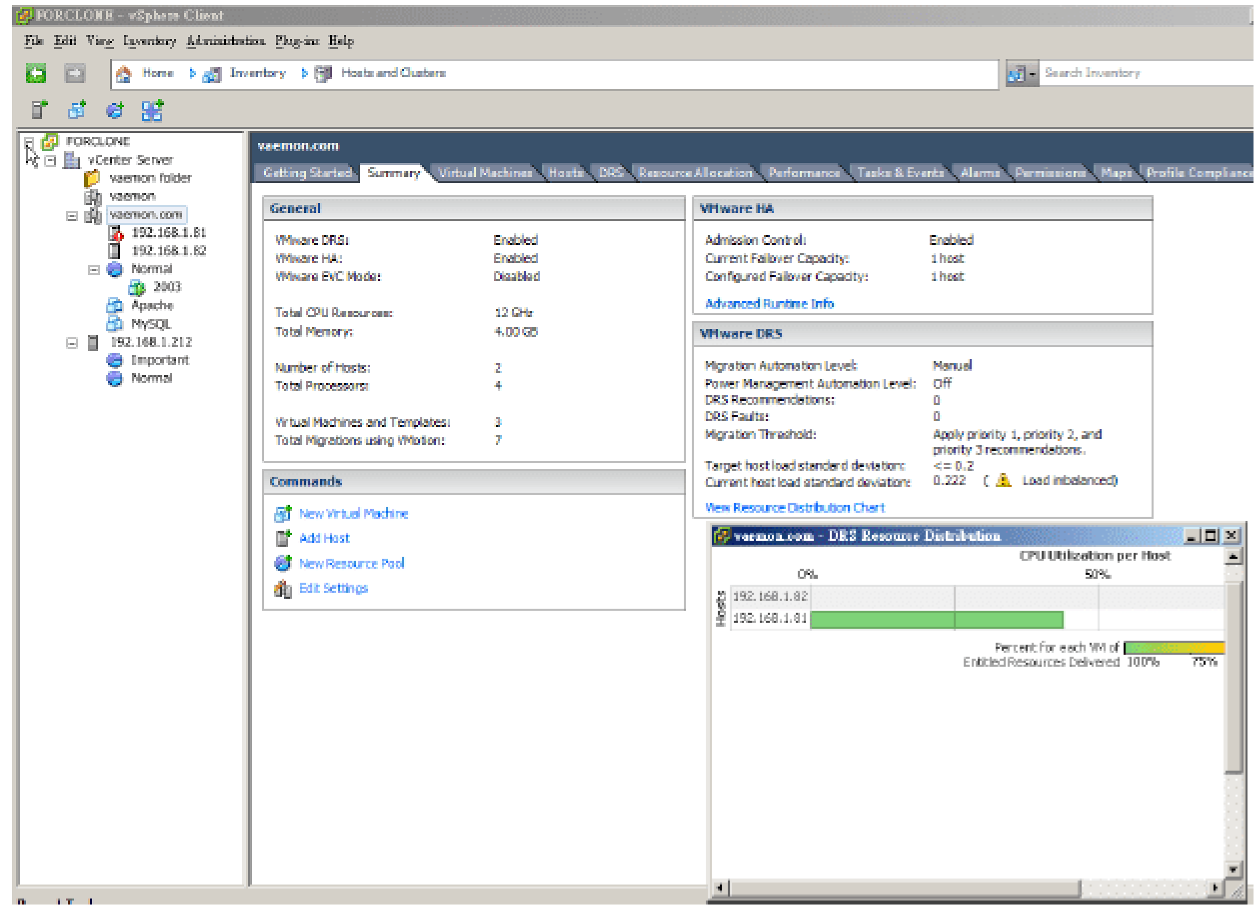


▲ 这是主机资源吃紧的提示



▲ 选择集群名称并且检视资源的使用

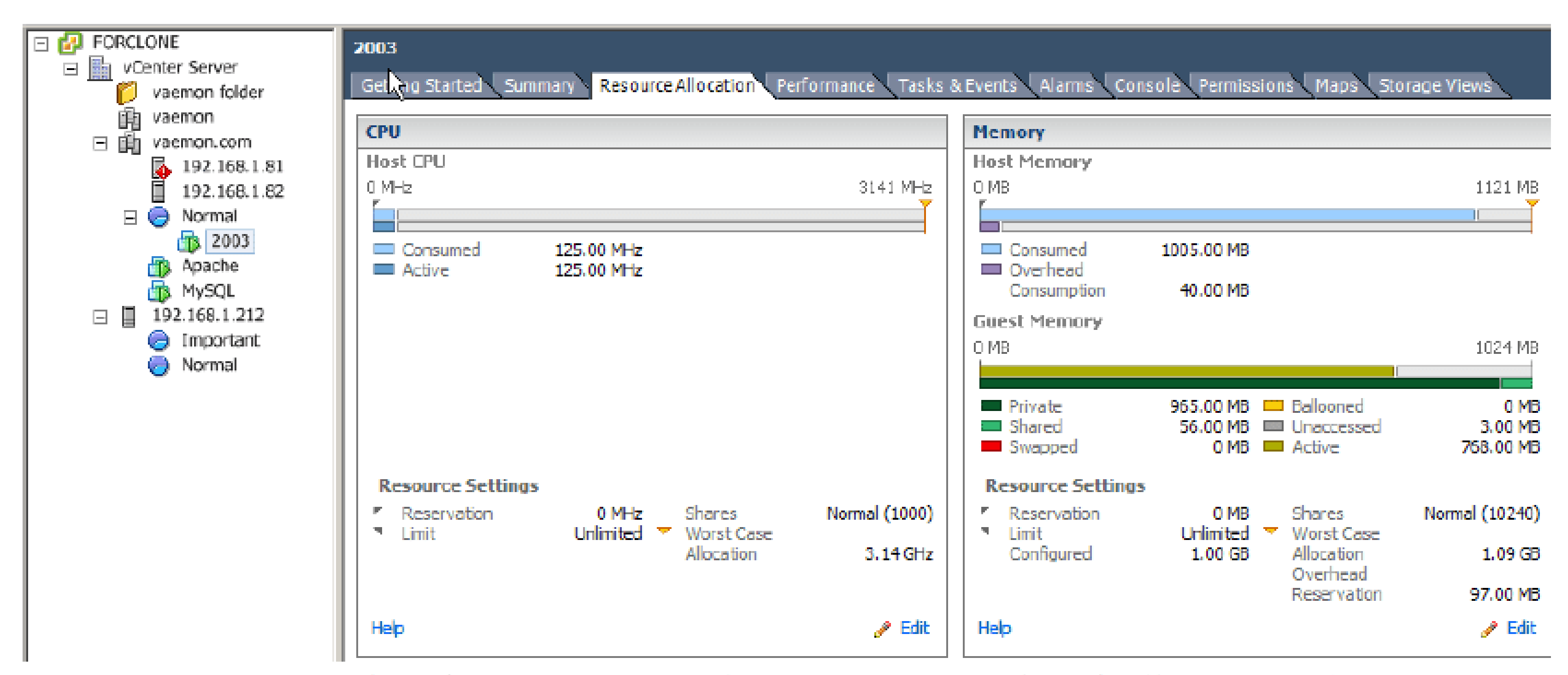
3. 在下图中，我们可以看到两台主机的使用情况，使用第一台 ESX 的 CPU 在 10%以下，也有在 10%～20%之间的，而内存则是在 10%～20%之间，最高到过 60%～70%。



▲　这是当前整个集群的资源使用状况

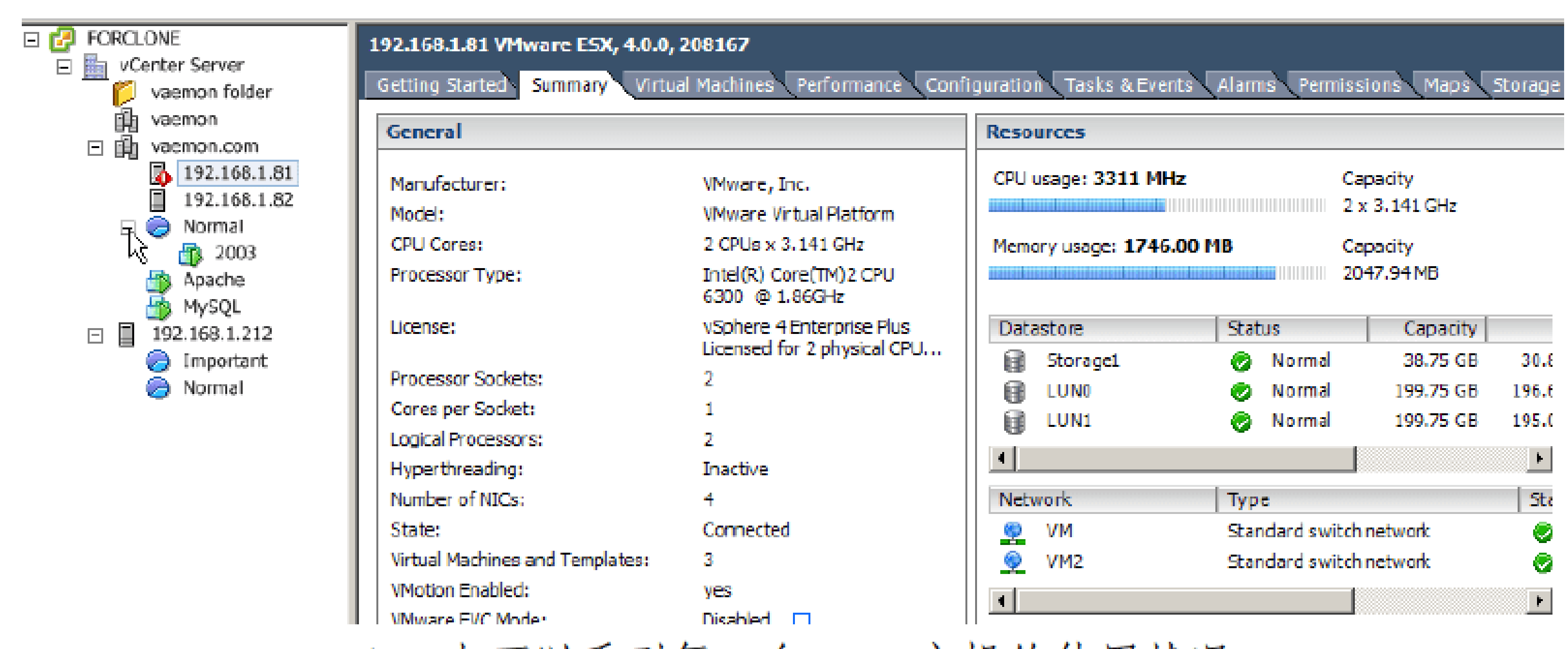
4. 当使用第二台主机时，可以看到整个 CPU 和内存都高到 90%以上，意味着系统资源曾经有过吃紧的情况。

5. 当我们选择 Virtual Machines 选项卡时，可以看到不同的虚拟机当前的 CPU 和内存使用状况。当系统资源占用太多时，DRS 会开始激活 VMotion 转移。



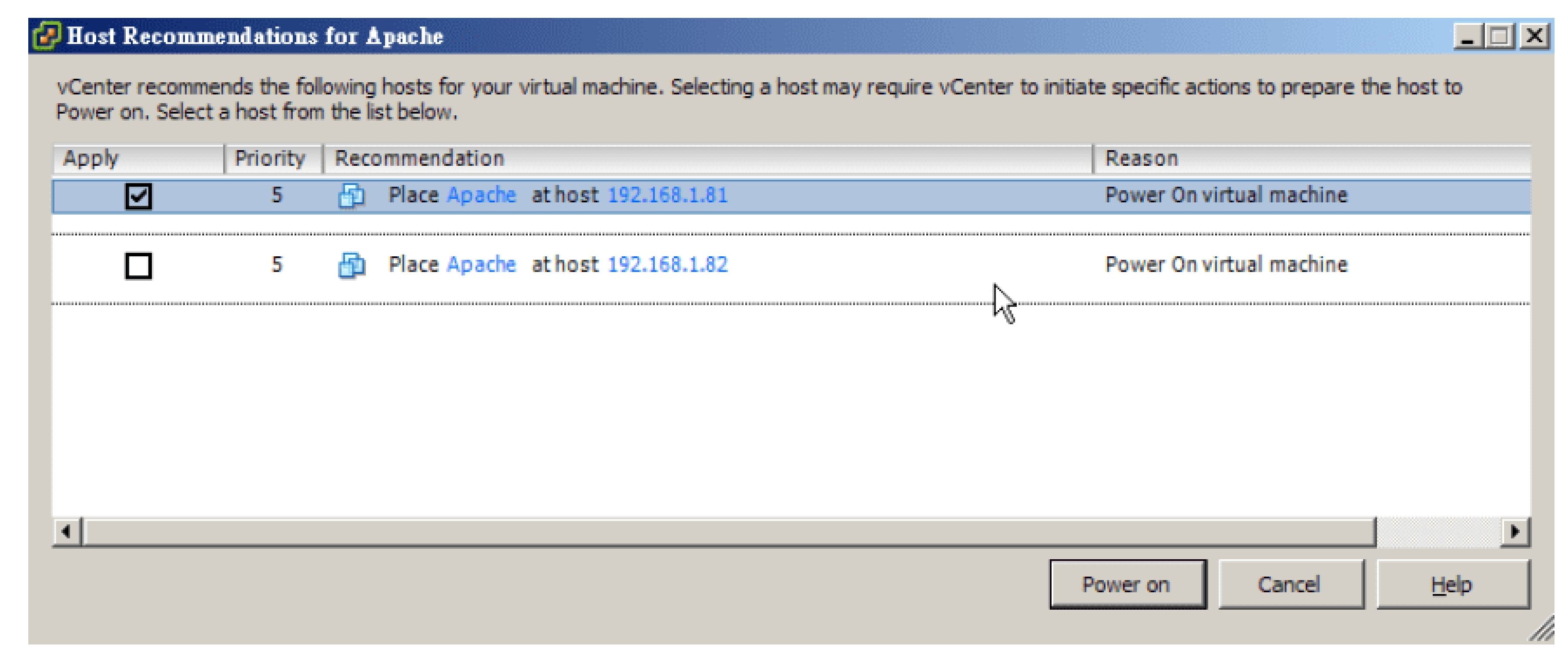
▲　选择虚拟机，会给出当前这个集群上的虚拟机使用资源状况

6. 进入 Hosts 选项卡，也可以看到当前两台 ESX 主机的资源应用情况。



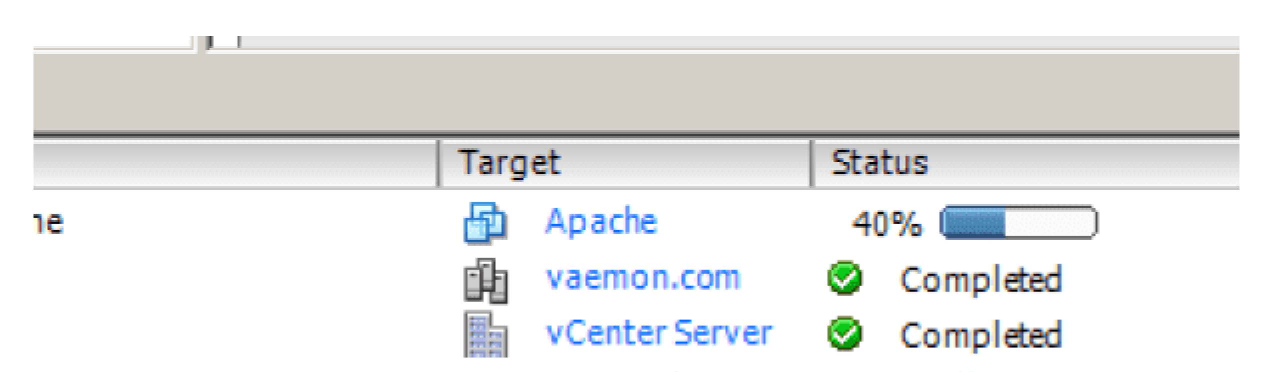
▲　也可以看到每一台 ESX 主机的使用状况

7. 当我们将 DRS 配置为半自动时，可以看到系统弹出建议了，DRS 此时建议将 2003 这台虚拟机移到 192.168.1.81，因为 192.168.1.82 的资源已经不够了。此时单击 Apply Recommendations 按钮将会激活 VMotion 把 2003 这台虚拟机移到 192.168.1.81 上。



▲ 当系统资源发生改变时，会生成新的建议

8. 在接受之后，就可以看到 DRS 正在进行 VMotion 的转移。



▲ 选择了接受建议，就会开始移转，当然使用 VMotion

根据笔者自身的经验，在说服客户投入 vSphere 架构的虚拟机应用时，大部分的客户最关注的热点，就是单台的物理服务器是否能应付众多操作系统的负担，让每一个在独立操作系统中的服务都能跑得平顺，并且充分应用所有的硬件资源。因此 DRS 对于 vSphere 来说，就是服务层级的 QoS，保证每一个虚拟机都能和物理机一样提供平顺且不中断的服务，因此在配合了 VMotion 之后，除非硬件数据本来就不足，vSphere 上的虚拟机，永远不会因为系统资源的不足而引起服务变慢甚至是停止的情况。

结 语

从本章的说明，相信读者们可以理解，在 vSphere 架构中，为了确保每一台虚拟机都能平稳地运行，集群、DRS、资源池、VMotion 等部件，是确保 vSphere 架构中服务平顺的重要基础，也难怪 VMware 特别将 DRS 作为整个产品线中的主力强打。我们在下一章，将落实 vSphere 永远不蓝屏平台：高可用性（HA）功能，让 vSphere 平台真正能提供 24×7 的不中断服务。

第 17 章 创建不会蓝屏的环境：vSphere 的高可用性（High Availability）

关键词：

- vSphere 中的 HA
- 运行 HA 时必须遍历的热点
- 运行 VMware HA 必须遍历的条目
- 理解 HA 的资源应用
- HA 的重新激活选项
- 进行 HA 集群的规划原则
- 隔离回馈的运作原理
- VMware HA 原理及步骤
- 使用 NIC Team 来落实心跳线的冗余
- 使用 VMware HA 功能

vSphere 能自动将虚拟机转到性能较佳的 ESX 主机，VMotion 也能在虚拟机不重新激活的情况下保证服务的继续，但这一切的前提都是 ESX 物理服务器不能蓝屏。如果 ESX 的物理服务器无预警关机了，那么运行在其上的虚拟机也就无预警地停止服务，这对企业造成的损害比物理还要严重。作为一个企业整体解决专案的领导厂家，vSphere 当然早就预防到这一点，让不同的 ESX 主机之间能形成集群而互相备援，这就是 vSphere 架构中最重要的一块拼图：高可用性 HA，我们将在本章来介绍这个 vSphere 架构中最让人安心的功能。

17.1 HA 和集群的关系

HA 这个功能并不是 vSphere 独有的，而是企业服务的最高指导原则。企业信息平台这个行业从几十年前就知道一个事实，就是用单独一台服务器来提供特定的服务是相当危险的。因此让多台计算机形成一个集群，并且让集群中其中一台计算机提供服务，集群中的其他计算机在平常的时候不提供服务而处于待命会话，当提供服务的主计算机（Primary Node）弹出问题时，其他的计算机在最短的时间内接手主计算机的服务，让系统的停机时间降到最短，使得整个服务能完整而连续，这也是 HA 的重要基础。

17.1.1 vSphere 中的 HA

在 vSphere 中，HA 是以一群 ESX 服务器为主的集群功能，主要的目的和微软的 Hyper-V Failover Cluster 完全一样，就是不希望一台物理服务器在蓝屏时，上面的虚拟机全部无法使用，让许多重要的服务（如邮箱、FTP、网页）并发全部停止，对企业的服务造成更大的损害。VMware HA 最重要的特色就是自动化、集群化和 VMotion。但 VMware 的 HA 和一般的 HA 一样需要多台服务器、独立的存储设备以及完善的网络规划才能落实，更和一般的 HA 集群会有潜在的停机时间及数据损失问题，但不管如何，VMware 的 HA 也充分运用到了虚拟机的优势和特色，我们就先来看看 VMware HA 的原理和热点。

当 VMware 集群激活 HA 时，会立即安装三个部件，分别是自动可用性管理员（Automatic Availability Manager，AAM）、VMap 和 vpxa，我们就来看看。

1. 自动可用性管理员

自动可用性管理员（AAM）是整个 VMware HA 的心脏，其维护一个内部数据库来表项集群中所有节点的消息。AAM 掌管了集群之间的心跳表项（Heartbeat），用来遍历哪一台节点是活的或已经蓝屏了。在集群中，每一台计算机都使用主控台通信端口来和其他的节点沟通。一般在配置 VMware 的 HA 时，会建议使用 NIC Team 两片以上的网卡来进行 HA 的心跳线以防止单点失误，我们会在下一节说明 NIC Team 的做法。

Device	Speed	Configured	vSwitch
82545EM Gigabit Ethernet Controller (Copper)			
vmnic1	1000 Full	Negotiate	vSwitch0
vmnic0	1000 Full	Negotiate	vSwitch0

▲ NIC Team 在 HA 上扮演了“心跳”线冗余的角色，需要两片网卡

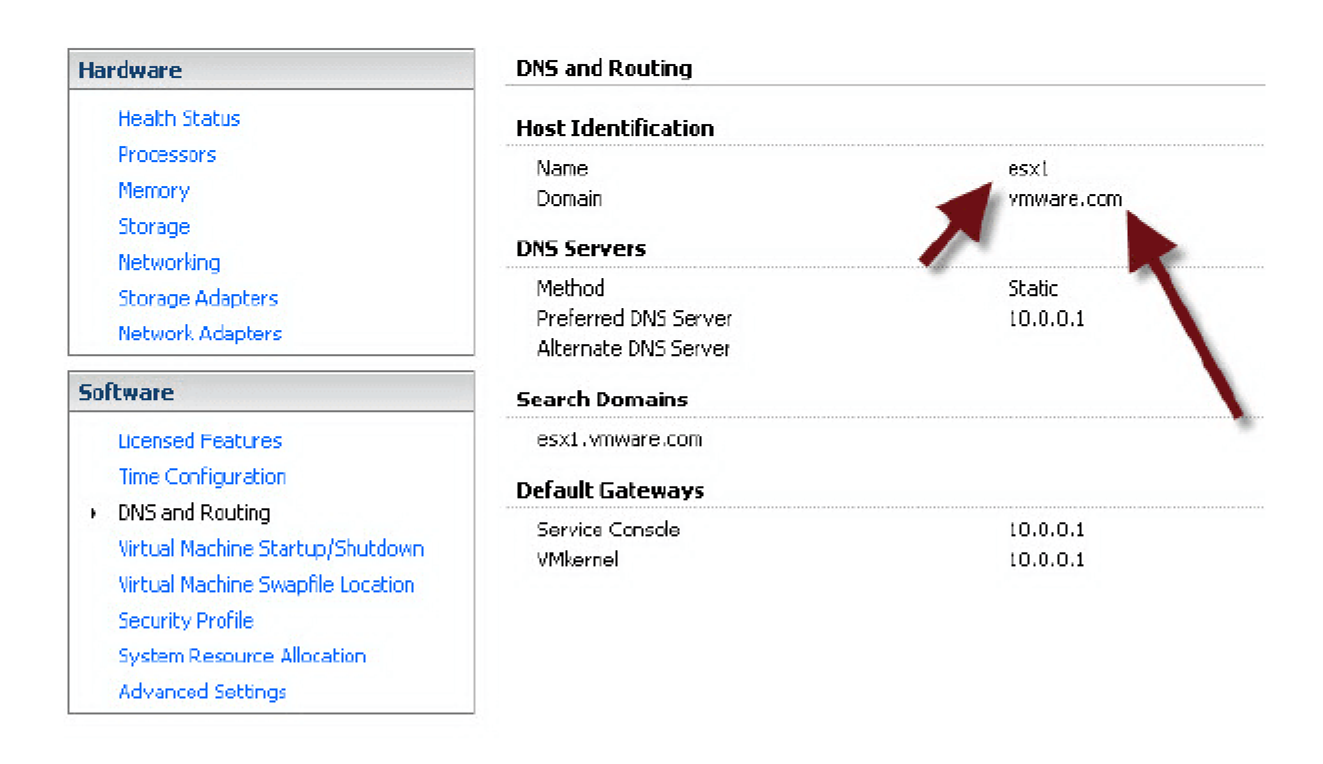
2. 必须使用主机名来管理 HA

由于 AAM 使用主机名而非 IP 来联络每一台计算机，因此在集群中的每一台计算机都必须有很完整的 DNS 及主机名配置。如果你从第 18 章开始就照着本书的步骤配置，那么你的 HA 环境也不会有太大的问题。

3. 运行 HA 时必须遍历的热点

- 在 vCenter 上能访问到每一台参加 HA 的计算机，如 IP 或 DNS 表项。

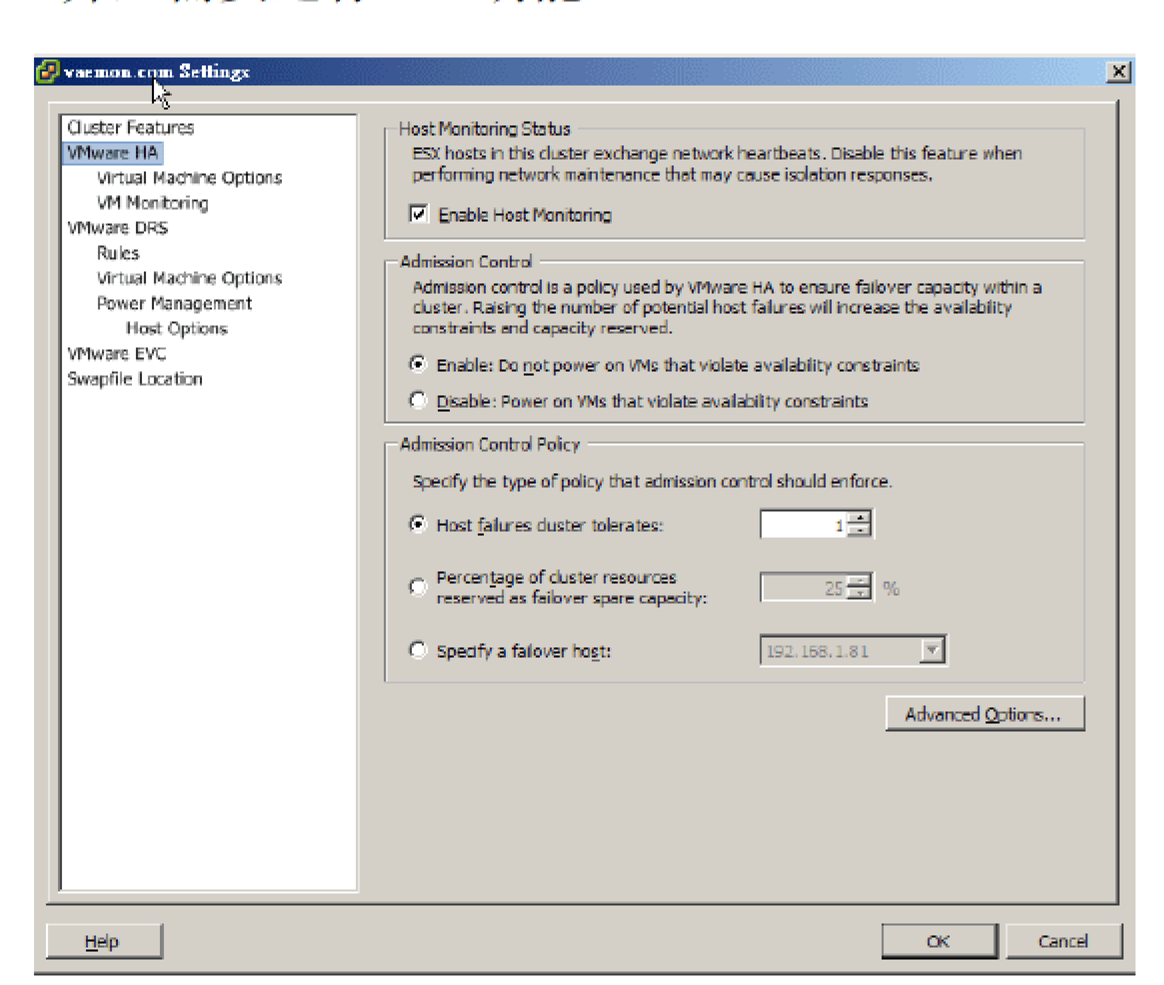
- vCenter 主机必须在所有节点引导前就常规运作
- 每一台节点计算机都必须能访问其他任意一台计算机节点，如 IP 或是主机名。



▲ 在 vCenter Server 中，最好以主机名而非 IP 来访问主机

4. 不再有主副节点

在上一版的 VI 中，HA 集群中的每一个节点都可以成为主节点（Primary Node）或是副节点（Secondary Node）。但在提供服务的时候，只有一台 ESX 服务器可以是主节点，其他所有的 ESX 服务器都是副节点。在 VMware 的 HA 中，第一台加入集群的 ESX 服务器被默认为主节点。但是在新版的 vSphere 中，主副节点的定义已经不再使用，而是使用 Cluster 的概念。在激活 HA 的 Cluster 中，所有的 ESX 会互相监视，你只要激活 Enable Host Monitoring，所有的 ESX 主机就会交换心跳（Heartbeat），并且需要运行 HA 功能。



▲ 激活 Host Monitoring 将会让 ESX 主机互相监视其他主机的“心跳”

5. 遍历 VMware HA 的配置值

运行 HA 和运行其他的 Failover Cluster 一样，必须要有多台 ESX 服务器，良好的网络设备及规划才行。一般来说，我们在运行 VMware 的 HA 之前，都会遍历几个条目。

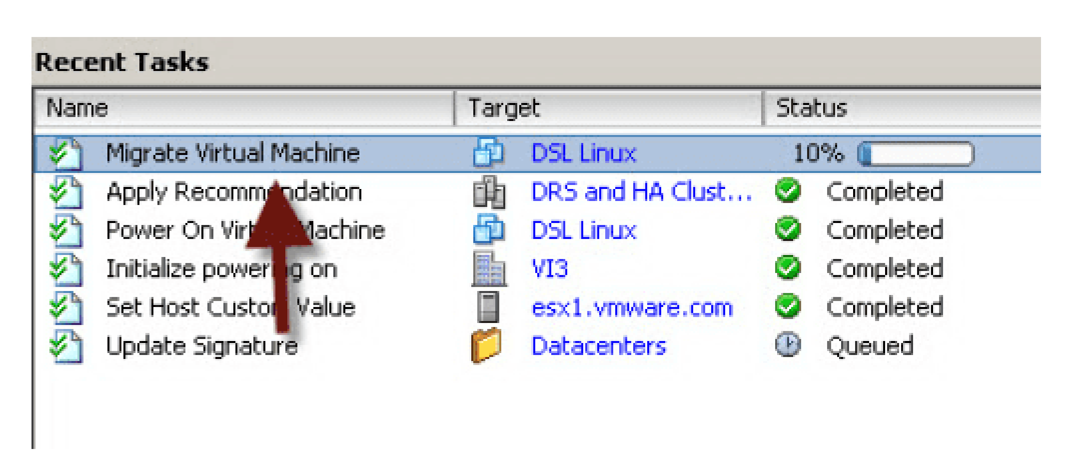
6. 运行 VMware HA 必须遍历的条目

- 所有 HA 集群中的节点都必须连上共同能访问到的网络存储，如 NFS、iSCSI 或是 FC

SAN。

- 所有 HA 集群中的节点都必须有完全一样的网络配置，如 vmnic、vSwitch、通信端口组。
- 所有 HA 集群中的节点都必须能访问到其他的节点。

基本上，只要你的计算机能运行 VMotion，运行 HA 就没问题了。举例来说，如果你的 HA 集群中有 3 台节点 A、B、C，你只要运行一次三个节点之间的 VMotion，没问题的话，VMware 的 HA 也大致确定了。



▲ 只要你能在集群中的计算机常规使用 VMotion，HA 大致也没问题了

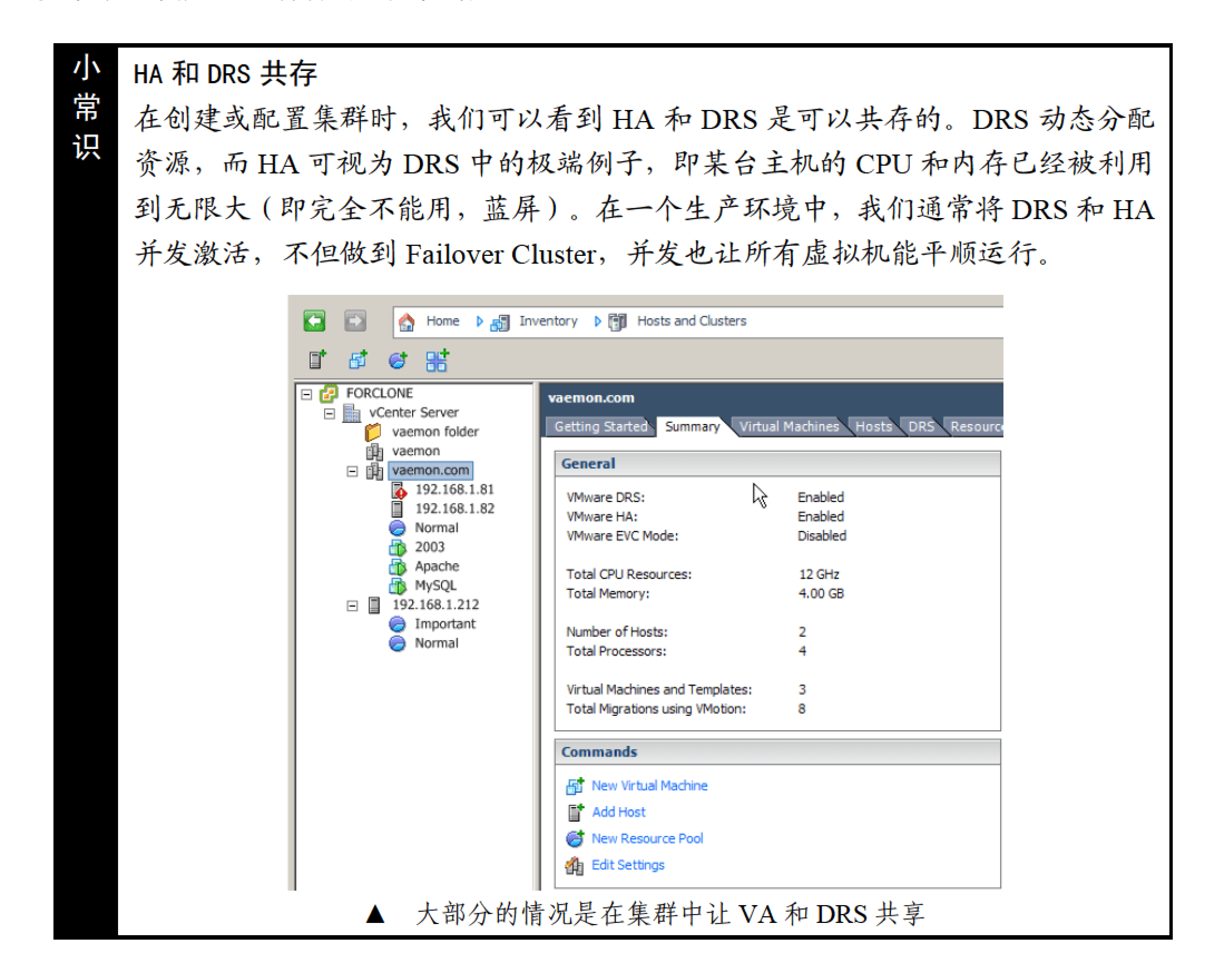
17.1.2 VMware HA 的基本选项

VMware HA 也是在 vCenter Server 中配置。在配置 VMware 中的 HA 集群和配置 DRS 的集群一样，只要在创建集群时，将 HA 激活即可。在激活了 HA 之后，就会弹出对应的 HA 参数，而这些参数就是决定 HA 活动的重要依据。

小常识

HA 和 DRS 共存

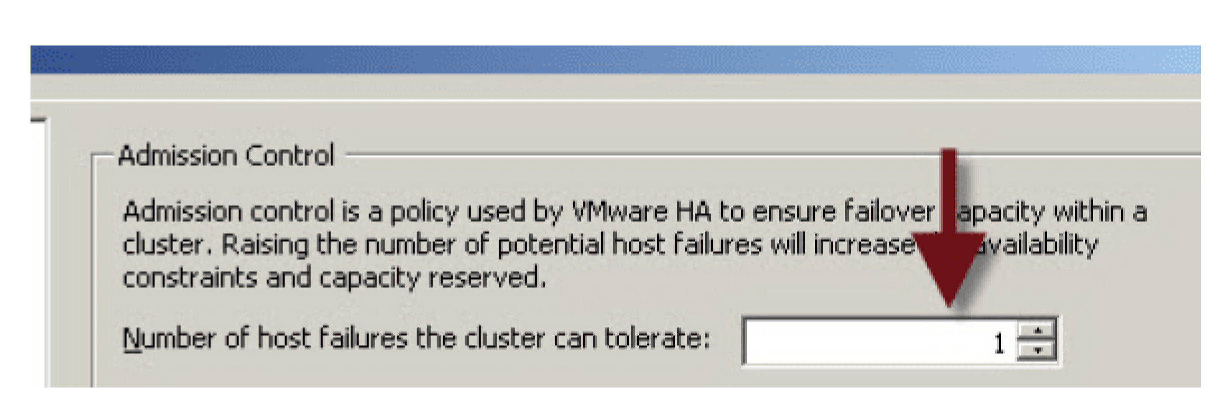
在创建或配置集群时，我们可以看到 HA 和 DRS 是可以共存的。DRS 动态分配资源，而 HA 可视为 DRS 中的极端例子，即某台主机的 CPU 和内存已经被利用到无限大（即完全不能用，蓝屏）。在一个生产环境中，我们通常将 DRS 和 HA 并发激活，不但做到 Failover Cluster，并发也让所有虚拟机能平顺运行。



▲ 大部分的情况是在集群中让 VA 和 DRS 共享

1. 配置 HA 的集群：可容忍主机蓝屏数量

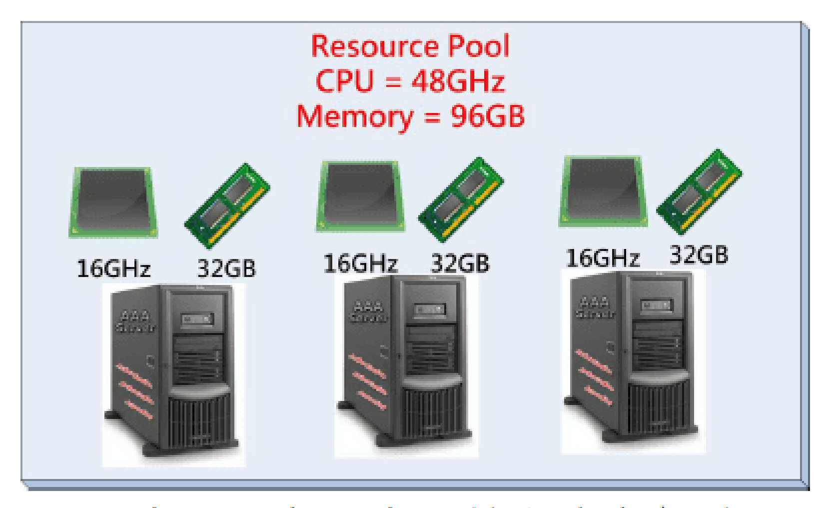
当我们创建 HA 集群时，最主要的参数就是配置 HA 的可承受的蓝屏主机数，允许总控以及虚拟机的单独选项。其中最重要的就是可承受的蓝屏主机数（Host failures allowed）。这个数字的意思常常被许多使用 VMware 的人误解了，正确的意思是“在这个 HA 集群中，你能忍受多少台主机并发蓝屏”。当然这个数字不是可以随便配置的，必须看整个集群中总资源，以及集群中所有计算机上运行的虚拟机总资源数。在理解这个数字之前，我们必须先理解一下整个 HA 集群中资源的计算。



▲ 这个数字通常是物理主机数的一半，但还是要看整体资源占用量

2. VA 集群必须精确计算资源

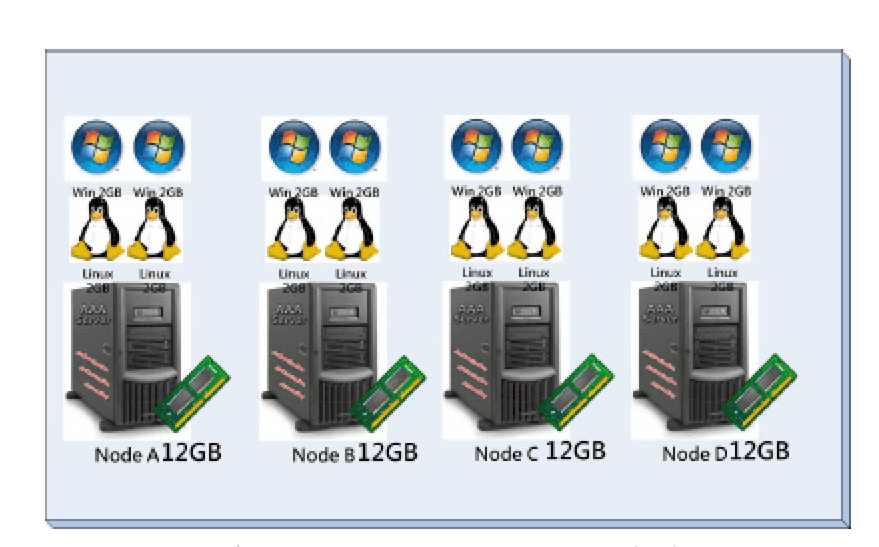
举例来说，如果有三台物理服务器分别使用了 2 个四核的 2.0GHz CPU、32GB 的内存，那么这个集群就聚合了 48GHz 的 CPU 资源和 96GB 的内存。当其中一台物理服务器蓝屏时，原来的总资源就剩下 32GHz 和 64GB。由于在 VMware 的 HA 中，剩下的两台物理服务器必须顶上去原来的服务，而这些物理服务器本身可能也有一些服务在运行，因此在设计 VMware 的 VA 架构时，必须先经过计算资源，才不会出现在 HA 激活时，蓝屏计算机上的虚拟机找不到可用资源的情况发生。



▲ 在 HA 中，资源是所有主机的和

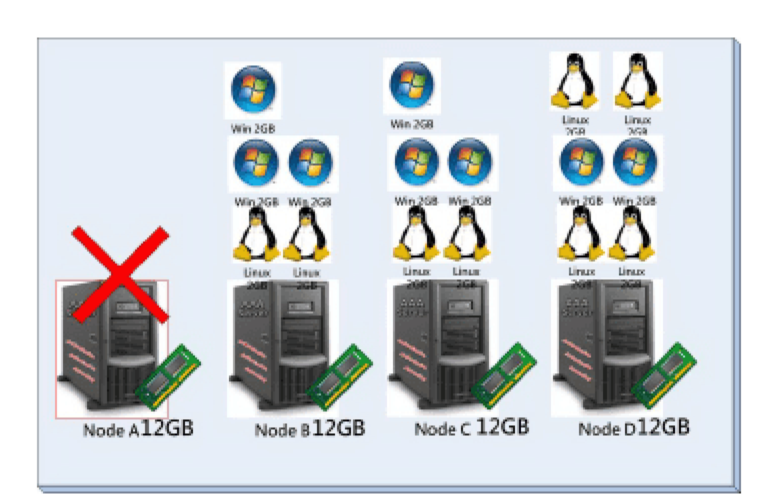
3. VA 集群资源计算实例：资源够用的情况

在 HA 的集群中，如果主节点蓝屏了，那么主节点上的虚拟机按照道理来说应该被分配到副节点上。但在虚拟机的环境中，这个情况并不是这么简单的。举例来说，如果在一个有四台节点的 HA 集群中，每一台 ESX 主机拥有 12GB 的内存，其中 A 节点是主节点，B、C、D 是副节点。



▲ 图中有四台主机 ABCD，A 为主节点，BCD 为副节点

在 A 节点上本来有四台虚拟机，一共用了 8GB，而 BCD 三个节点也各有四个虚拟机，也用了 8GB，再加上本身的 ESX 系统需要用掉 1GB，因此 BCD 节点各空出 3GB 的内存，一共 9GB 的空闲。当 A 节点蓝屏时，其上的四个虚拟机必须到剩下的 BCD 三个 ESX 主机上运行，由于在 A 节点上的所有虚拟机共需要 8GB 的内存，而 BCD 能提供 9GB 的内存，因此资源是足够的。

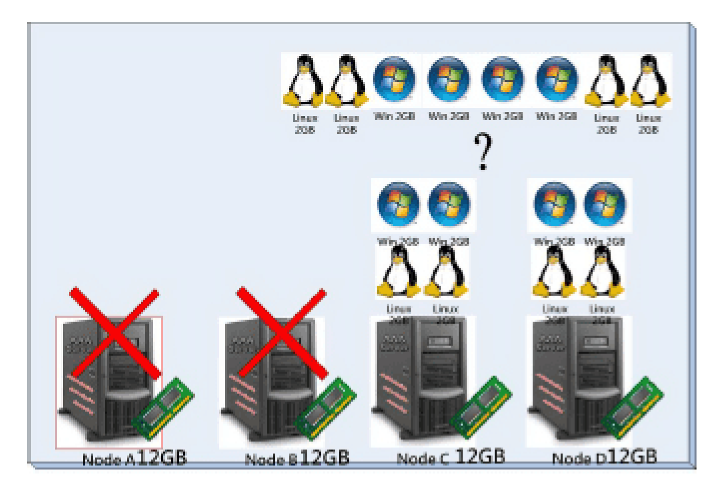


▲ 当节点 A 蓝屏时，剩下的 BCD 资源是可以分给 A 上的虚拟机的

4. VA 集群资源计算：资源不够用的情况

进入选项之后，首先要将 Enable Host Monitoring 选中，HA 功能才能常规使用。

上面的例子，如果在前面的 HA 集群中，B 节点也并发蓝屏时，此时就会弹出只剩下 CD 两个节点能用的情况。由于 CD 两个节点只有 6GB 的可用内存，而 AB 的蓝屏将会有 8 个虚拟机需要 16GB 的内存，此时会有 10GB 内存的不足。在这种情况下，虚拟机是否要从 AB 移到 CD 并且激活呢？这个动作是否运行，就要看我们配置的“可容忍主机蓝屏数”和允许总控的参数了。



▲ 当节点 B 也蓝屏时，CD 的资源就不够用了

5. 允许总控的配置

在 Admission Control 的类别中，在理解了资源不够的情况之下，我们才能进行第二项配置。第二项配置称为“允许总控”，其中有两个选项。第一个选项为“当虚拟机所需要的资源超过可用资源时不激活虚拟机”（Do not power on virtual machines if they violate availability constraints）。第二个选项则是“就算虚拟机所要的资源超过了可用资源也要激活虚拟机”（Allow virtual machines to powered on even if they violate availability constraints）。

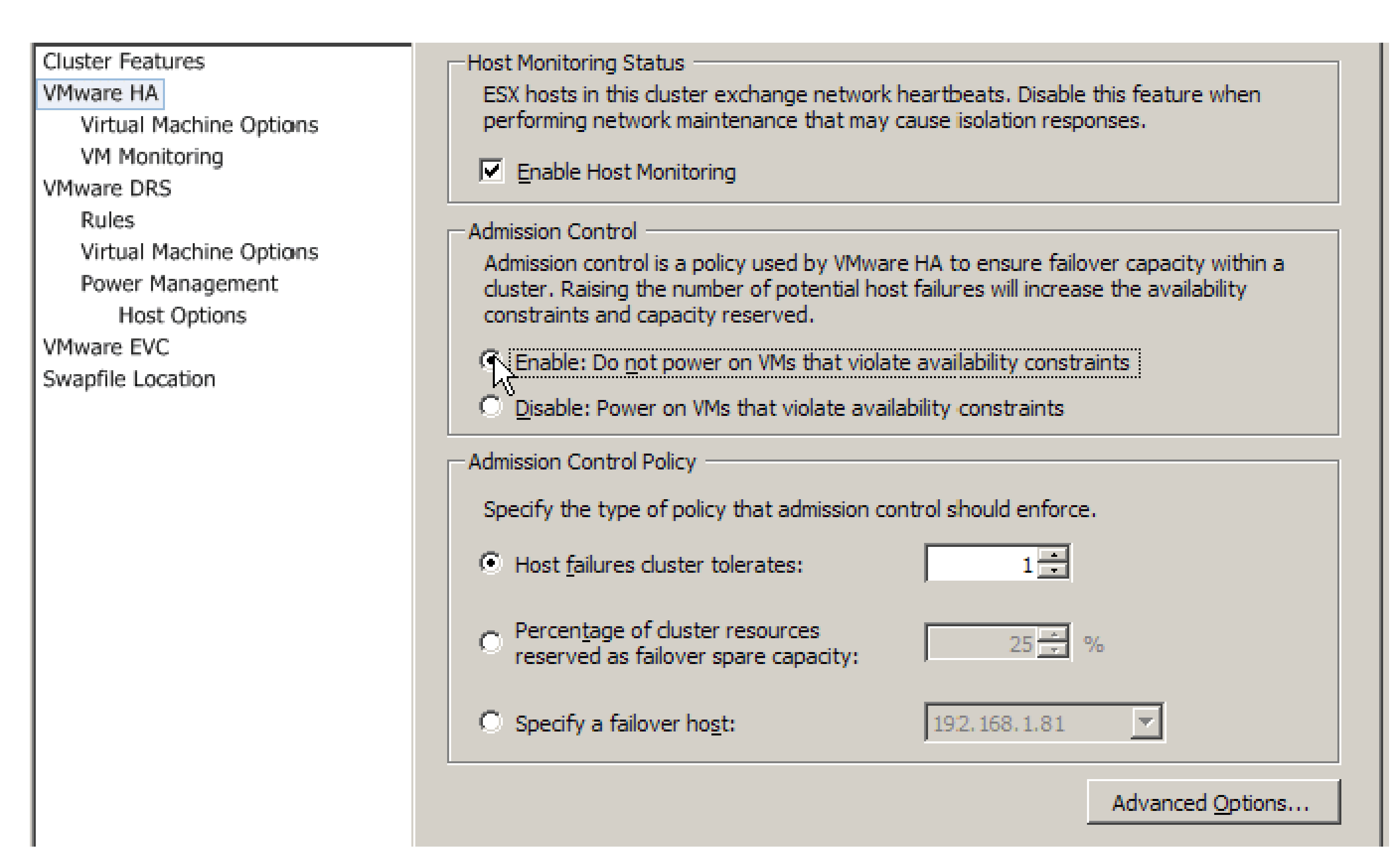
如果我们选择了第一个选项，会有可能在蓝屏发生之后，许多虚拟机都不会激活了，因为资源不够了。选择第二个选项会让虚拟机在副节点上激活，但由于资源的不足，因此激活后的虚拟机很可能无法平顺使用，这时就算 DRS 激活也无济于事，因为这时已经碰到了整体资源不够的最坏情况了（Worst Case）。

最后在第三个选项（Admission Control Policy）中，有三个配置，说明如下。

6. Admission Control Policy 的配置

- Host Failure cluster tolerates：整个 HA 允许蓝屏的数量。

- percentage of cluster resources reserved as failover spare capacity：留下来做 HA 的整个 Cluster 的资源比例。
- Specify failover host：用来作为 failover 的 ESX 主机。



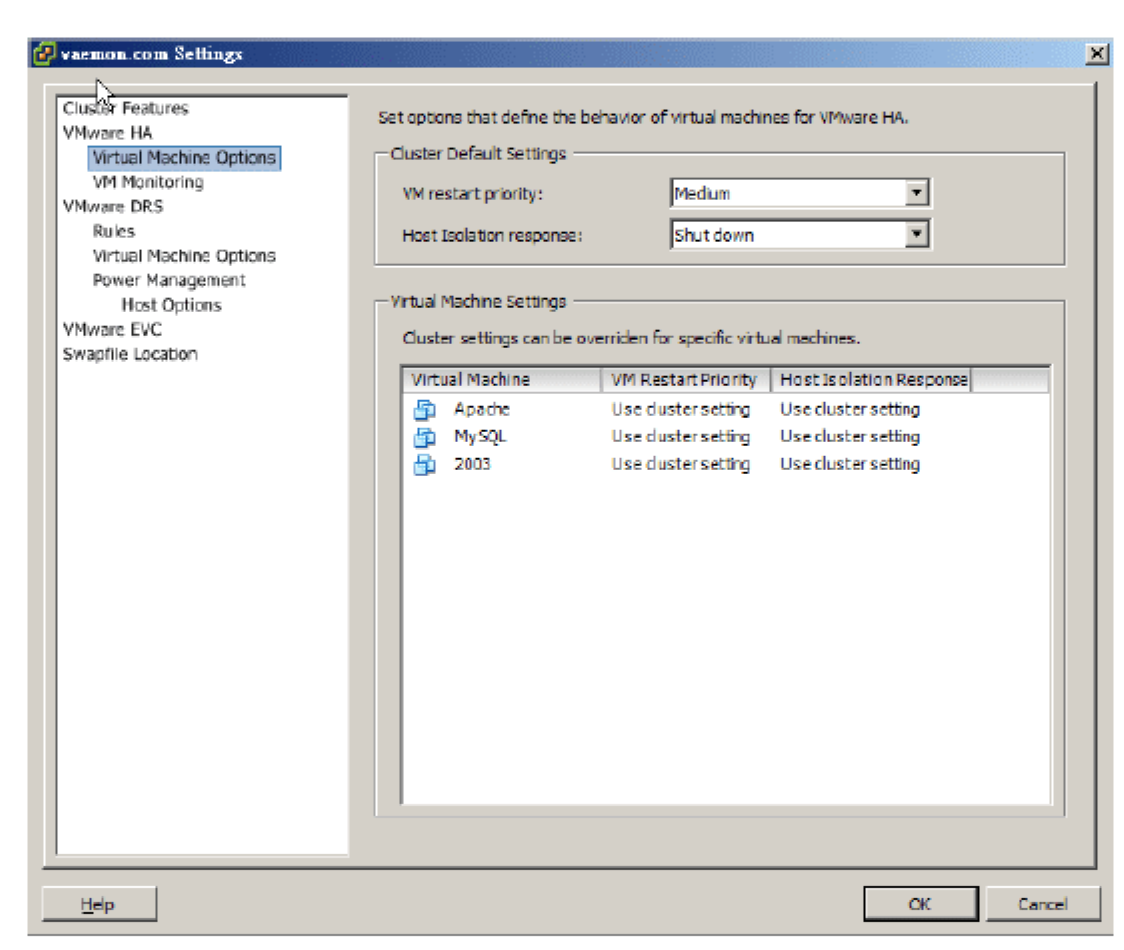
▲ HA 的三个大配置项

17.1.3 HA 的 VM 重新激活选项

在 vCenter 的 HA 中，有一个 Virtual Machine Options。在理解了资源不够的情况之后，我们又碰到和 DRS 一样的问题了。当资源不够时，有些虚拟机是不会激活的。举例来说，当 ESX 主机 A 蓝屏时，其上的四个虚拟机必须到 B 和 C 两台 ESX 服务器上激活。但 B 和 C 的资源只够两个虚拟机激活，那 HA 会在 BC 两台 ESX 主机上激活哪两台虚拟机呢？此时就是配置激活优先序的时候了。

1. 激活优先序

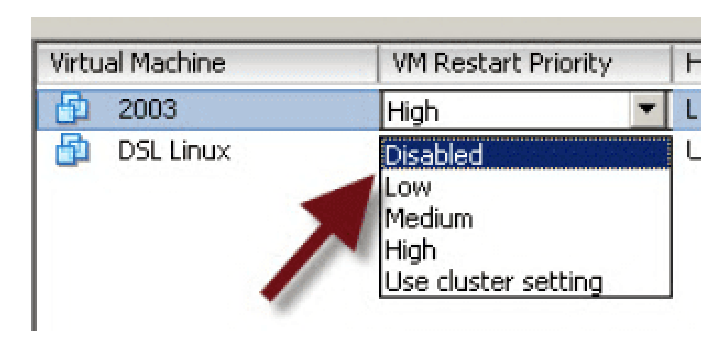
在 VMware 的 HA 中可以配置单个虚拟机的激活优先序。只要在你的 HA 集群中来进行配置即可。在 HA 集群中，虚拟机的激活优先序分为高、中、低和不激活四种。高、中、低从字面上的意义就知道了。而不激活意义，就是在这台虚拟机所处的 ESX 蓝屏之后，配置为不激活的虚拟机在资源吃紧时永远不会激活的。



▲ 可以在 HA 的选项中配置

2. 配置优先序之外

事实上，配置优先序也会弹出高优先序虚拟机无法激活的时候。举例来说，当资源只够 4 台虚拟机在没有蓝屏的节点上激活时，但如果你的虚拟机设为高优先序的有 5 台，并且配置原则为资源不够时就不激活，那么还是会有一台虚拟机无法激活，这时系统又必须伤脑筋是哪一台不激活，这种情况是 VMware 的 HA 也无法解决的时机。



▲ 也可以针对单个虚拟机来配置

3. 做好万全的规划

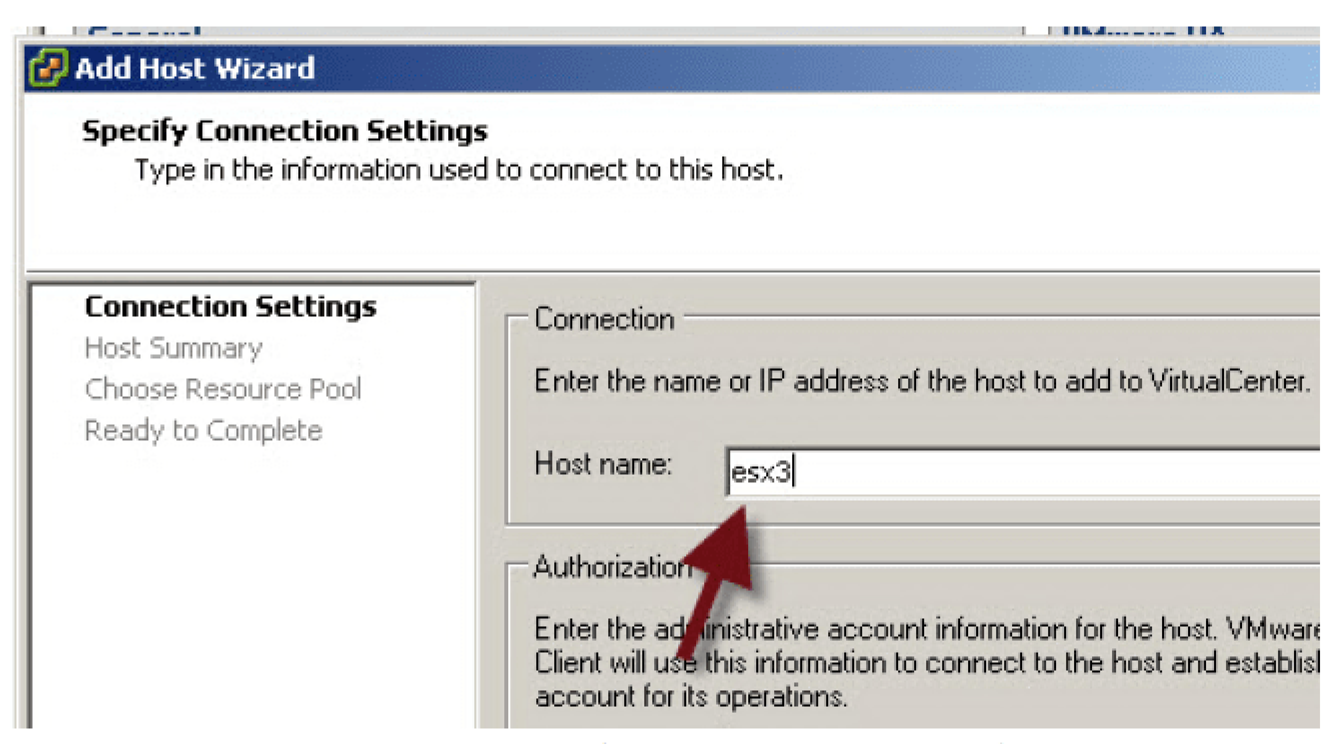
VMware HA 并不是全能的，一个完善规划的 vSphere 环境才是最重要的。在进行 VA 规划时，最重要的就是不要过度使用资源，下面就是规划时的原则。

4. 进行 HA 集群的规划原则

内存不要用到恰好。举例来说，每一台 ESX 主机在运行完所有的虚拟机之后，还能剩下一半的内存，如 HA 中有四台主机，每一台都运行 2 台虚拟机共享掉 4GB 的内存，那么机器最好准备 8GB 的内存，如果可容忍蓝屏的计算机数是在 2 台时。

配置能容忍蓝屏主机数时，HA 集群中的副节点要必须一样多。举例来说，能容忍蓝屏的主机数为 2 时，剩下的副节点就要是 2 台，HA 集群的总数必须是 4 台。

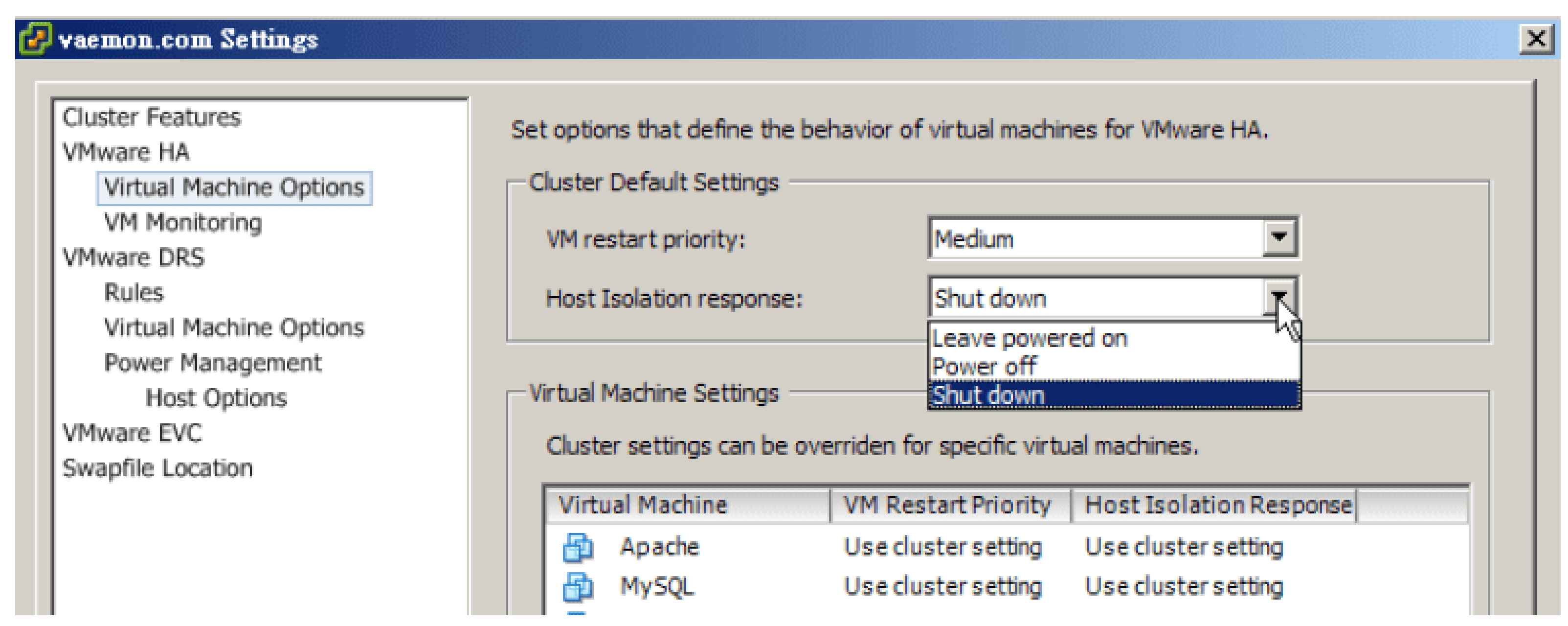
在系统资源充足的情况下，选择都激活虚拟机的选项，就算激活后的资源不如蓝屏前，但至少所有的服务还能常规使用。



▲ 在加入主机时，尽量用短名称

17.1.4 理解 HA 隔离回馈是什么

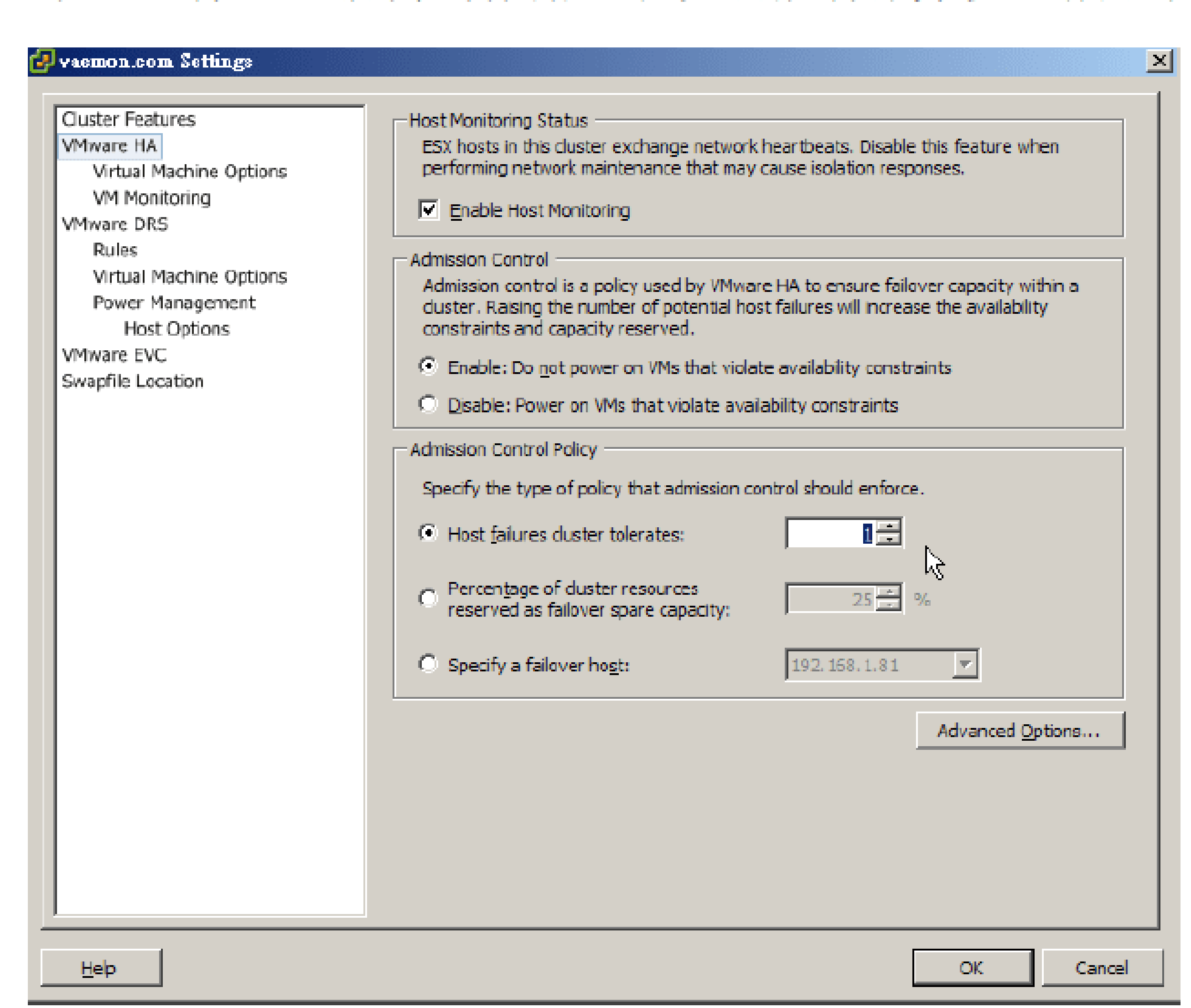
VMware HA 最主要的目的就是防止单个 ESX 服务器出错，造成其上所有的虚拟机都无法运行，因此在规划整个 VMware HA 的环境时，我们会尽量完备所有的设备以防止单点失误的发生。然而在网络的拓扑的设计上，常常因为规划不良而引起 HA 的错误或是冲突。我们就来看看最常见的例子。



▲ 隔离回馈是嗅探网络的手段，和主机本身无太大关系

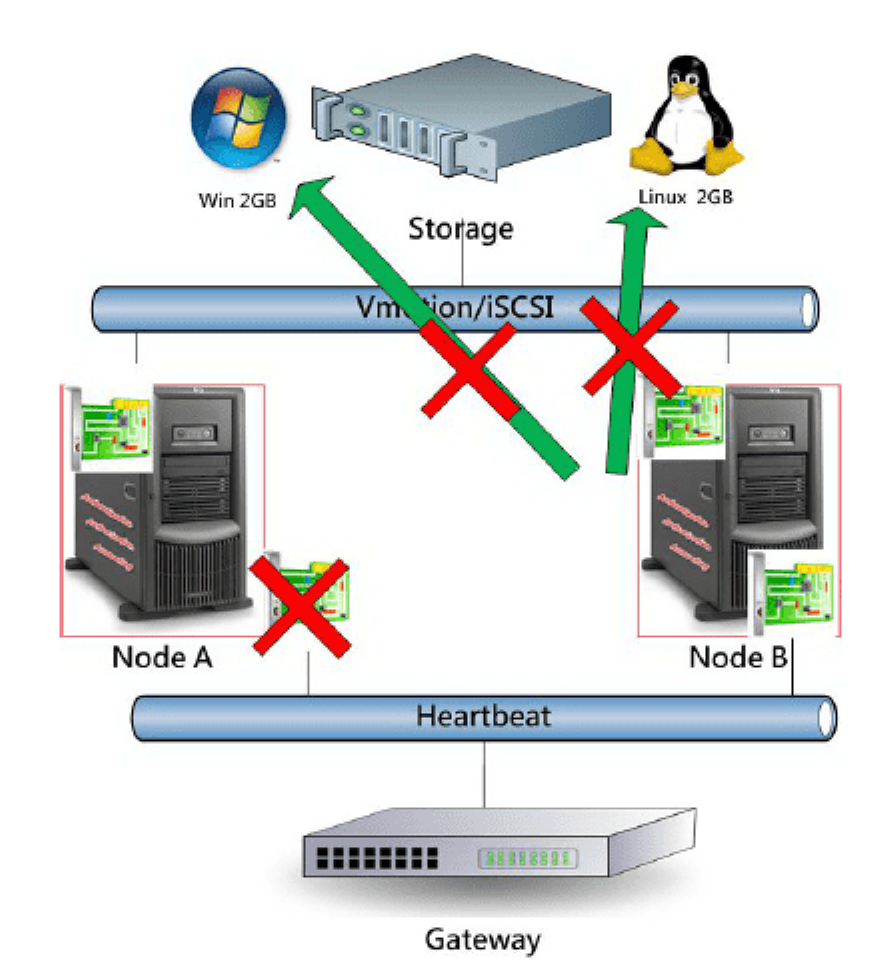
1. 重新激活或是不重新激活虚拟机？

举例来说，在一个有两台节点的 HA 集群中，某一台计算机 A 的心跳网络和虚拟机通信端口组是同一个网卡。当计算机 A 的心跳网卡突然停止了，但是计算机 A 主机本身并没有蓝屏，也能联机到存储设备上。此时主节点上的虚拟机因为使用同一片网卡，因此虚拟机也无法常规上网了。这个时候，VMware HA 会视这些虚拟机已经无法提供服务，会激活 HA 机制；计算机 B 嗅探到计算机 A 的心跳已经停止，就会到存储网卡设法激活计算机 A 所总控的虚拟机。



▲ 计算机 A 的网络故障，但本身没有故障

但问题来了，计算机 A 并没有蓝屏，只是心跳网络停止，其上的虚拟机还是常规运行（只是无法连上网络），会将存储设备上的 VMDK 或 VMX 文件锁定，因此计算机 B 到存储设备上就无法激活节点 A 上的所有计算机。这时计算机 B 就面临一个虚拟机死锁（Dead Lock）的问题。既然计算机 A 已经无法提供虚拟机常规运行的环境了（虚拟机通信端口组坏了），那计算机 B 又不能重新激活计算机 A 上的虚拟机（文件死锁），那么 VMware HA 不是就白做了？



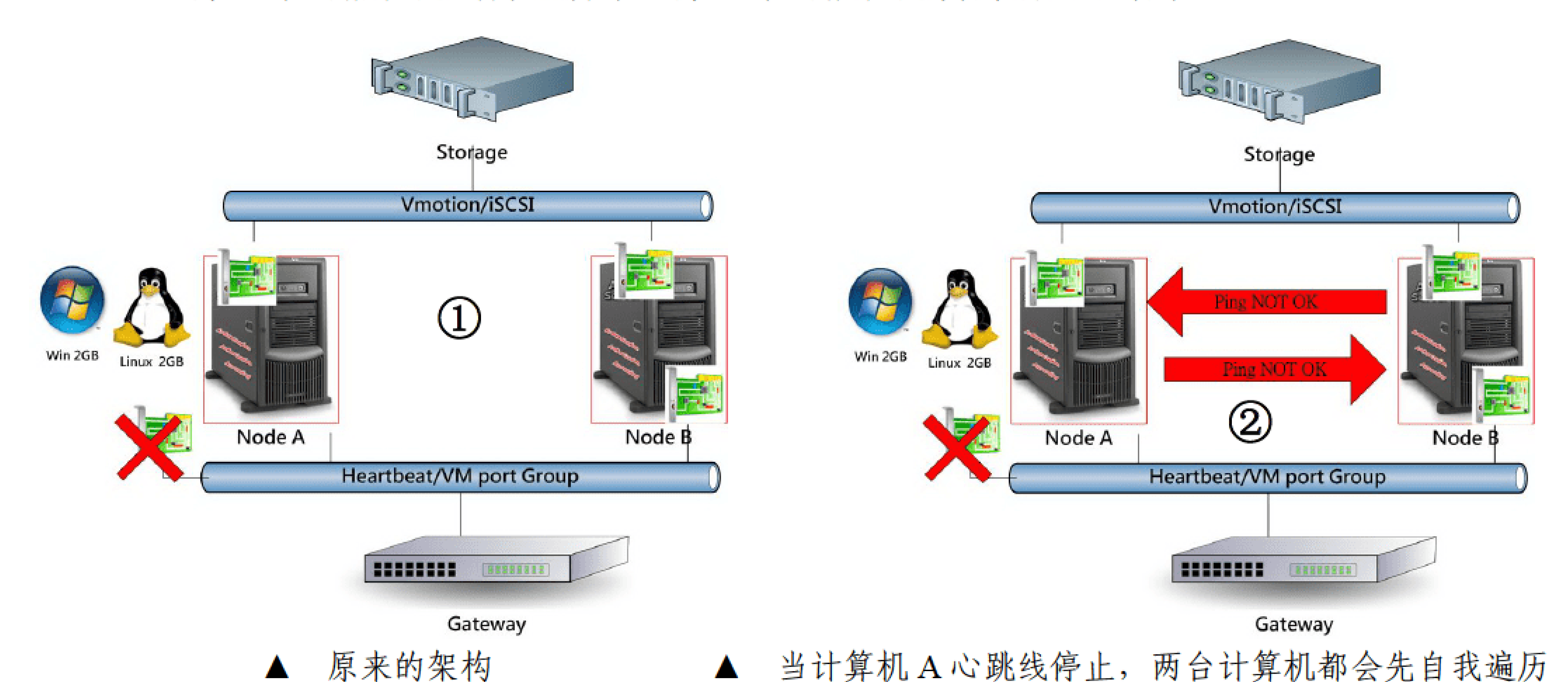
▲ 当计算机 A 没有蓝屏，又可以连上存储网络时，计算机 B 无法接管虚拟机

2. VMware HA 的隔离回馈

VMware HA 在设计时就已经想到这一点了，因此我们在配置 HA 的参数时，也可以顺便体会 VMware HA 在高可用性上的面面俱到。VMware HA 提出了一个所谓隔离回馈（Isolation Response）的 IP 位置，让失去心跳回馈的节点能对自身运行的虚拟机进行解锁的动作。

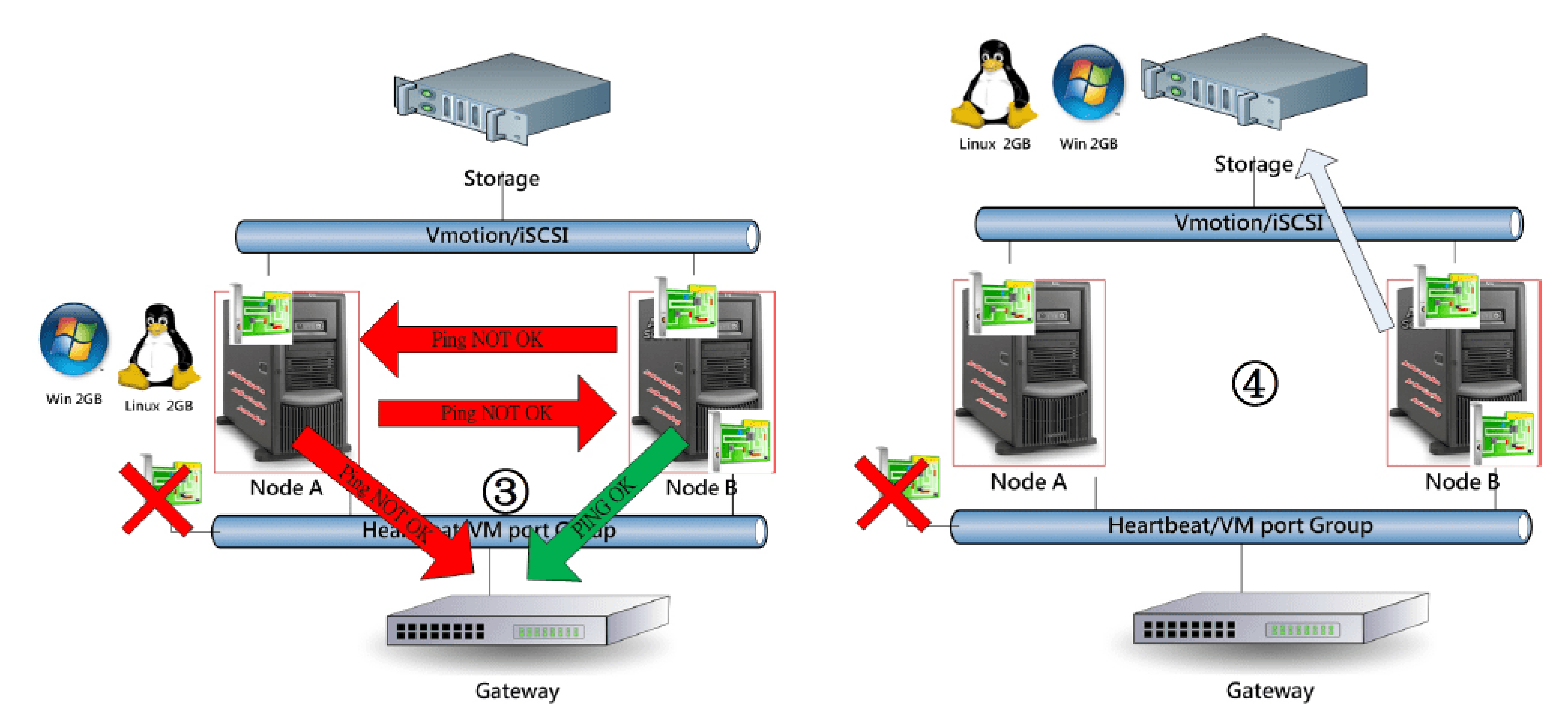
▶ 隔离回馈的运作原理

1. 当心跳网络和虚拟机通信端口组的网络位于同一张网卡或虚拟交换机上时，如果我们将这片网卡的网线拔掉。
2. 此时每一台节点都会发现有一台计算机已经心跳停止了。ping 不到对方有两个可能性，第一个可能性是对方蓝屏了，第二个可能性是自身的网络断了。



▲ 原来的架构　　▲ 当计算机 A 心跳线停止，两台计算机都会先自我遍历

3. 当 ping 不到对方 30 秒之后，每一个计算机会开始 ping 所谓的隔离回馈 IP。
4. 隔离回馈 IP 默认值是主控台通信端口的 Gateway，因为如果能 ping 得到自身的 Gateway，意味着主控台是常规的，此时如果计算机 B 能 ping 得到隔离回馈 IP，意味着自身这个节点是常规的，换句话说，弹出的故障是别台计算机，因此这个计算机 B 就会准备去找出坏掉的计算机，并且准备激活其上的虚拟机。

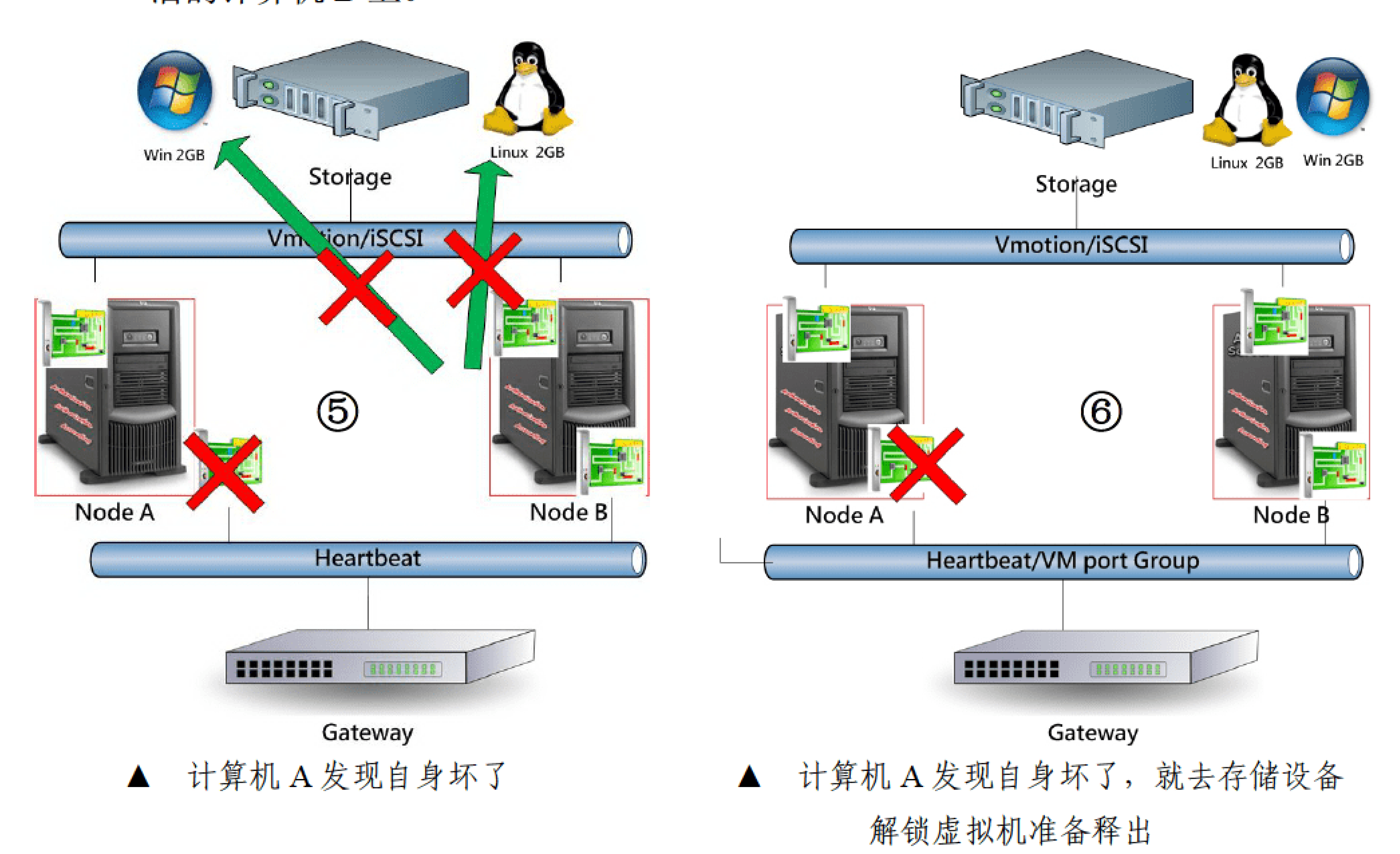


▲ 一般隔离回馈 IP 就是网关的 IP

▲ 计算机 B 发现自身没事，准备去存储设备接管 A 上的虚拟机

5. 如果计算机 A 在 30 秒之后发现 ping 一直没有连通，就会去 ping 隔离回馈 IP，结果发现隔离回馈 IP 也无法 ping 通。

6. 此时计算机 A 可以确定自身的心跳网络和虚拟机通信端口组已经故障，因此就会将自身所承载的虚拟机全部关机或是解锁，利用 VMotion 将这些虚拟机全部移动到还存活的计算机 B 上。



▲ 计算机 A 发现自身坏了

▲ 计算机 A 发现自身坏了，就去存储设备解锁虚拟机准备释出

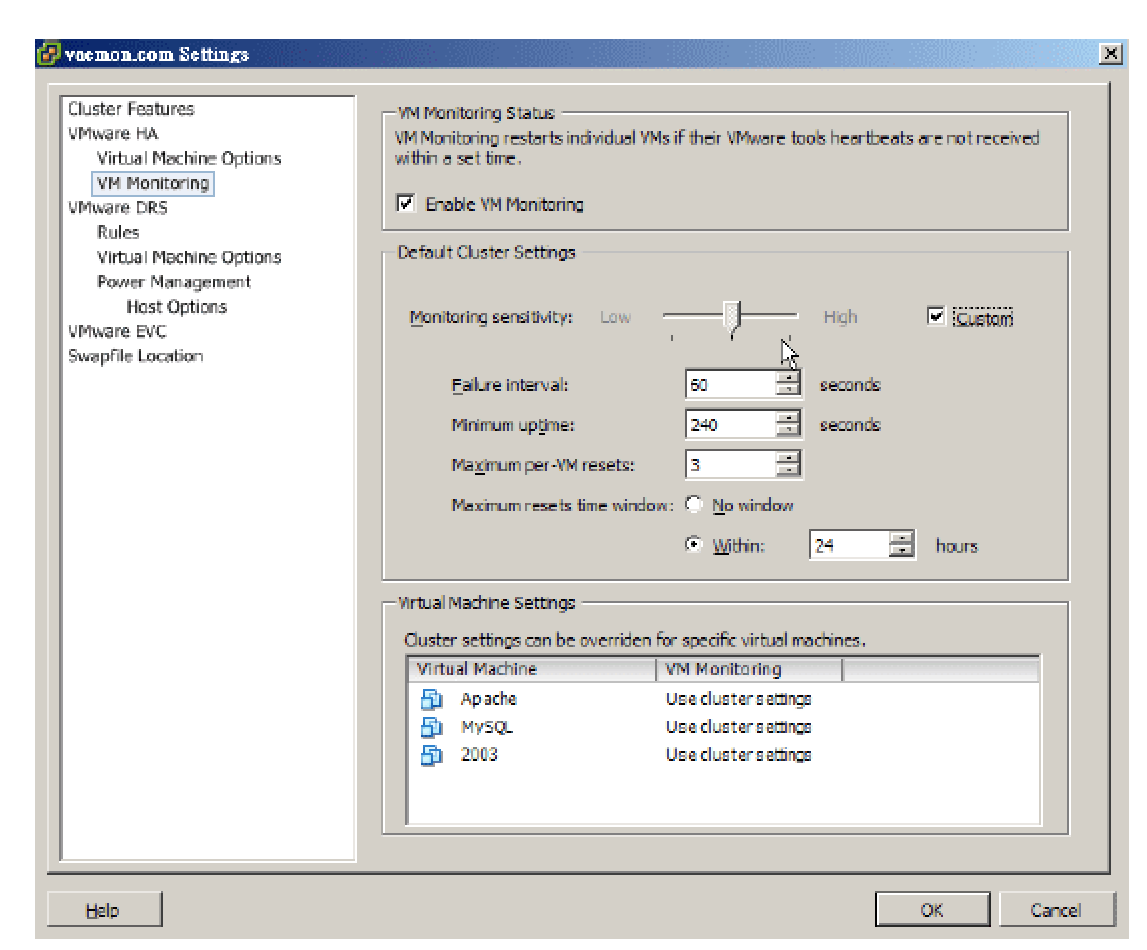
4. 理解 VM Monitoring 的选项

当选择 VM Monitoring 选项之后，会有几个重要的配置。这里是 HA 和 VM 之间的关系配置，而非 HA 和物理机之间的配置。由于 VM 是否在其他的 ESX 上重新激活，关系着整个系统的稳定性，因此这里的配置十分重要，我们就来看看。

5. VM Monitoring 配置项

- Enable VM Monitoring：这个选项会在监测不到 VM 的心跳时，重新激活 VM。
- Default Cluster Settings：这里让 HA 决定是否要重启 VM。在 LOW 的时候，每两分钟遍历一次，如果没有心跳就会重新激活。每七天会在前三次失败后重启。
- Medium：每 60 秒遍历一次，每 24 小时会在三次失败后重新 VM。
- High：每 30 秒遍历一次，每 24 小时会在三次失败后重新 VM。
- 自定义：自定义重启 VM 的方式。

在上面的自定义项中，你可以选择遍历 VM 失败的时间间隔、最小的存活时间遍历、最大的重启 VM 次数，以及在多少时间间隔内的最大重启次数等。



▲ 图为自定义的画面

6. 良好的 HA 网络设计

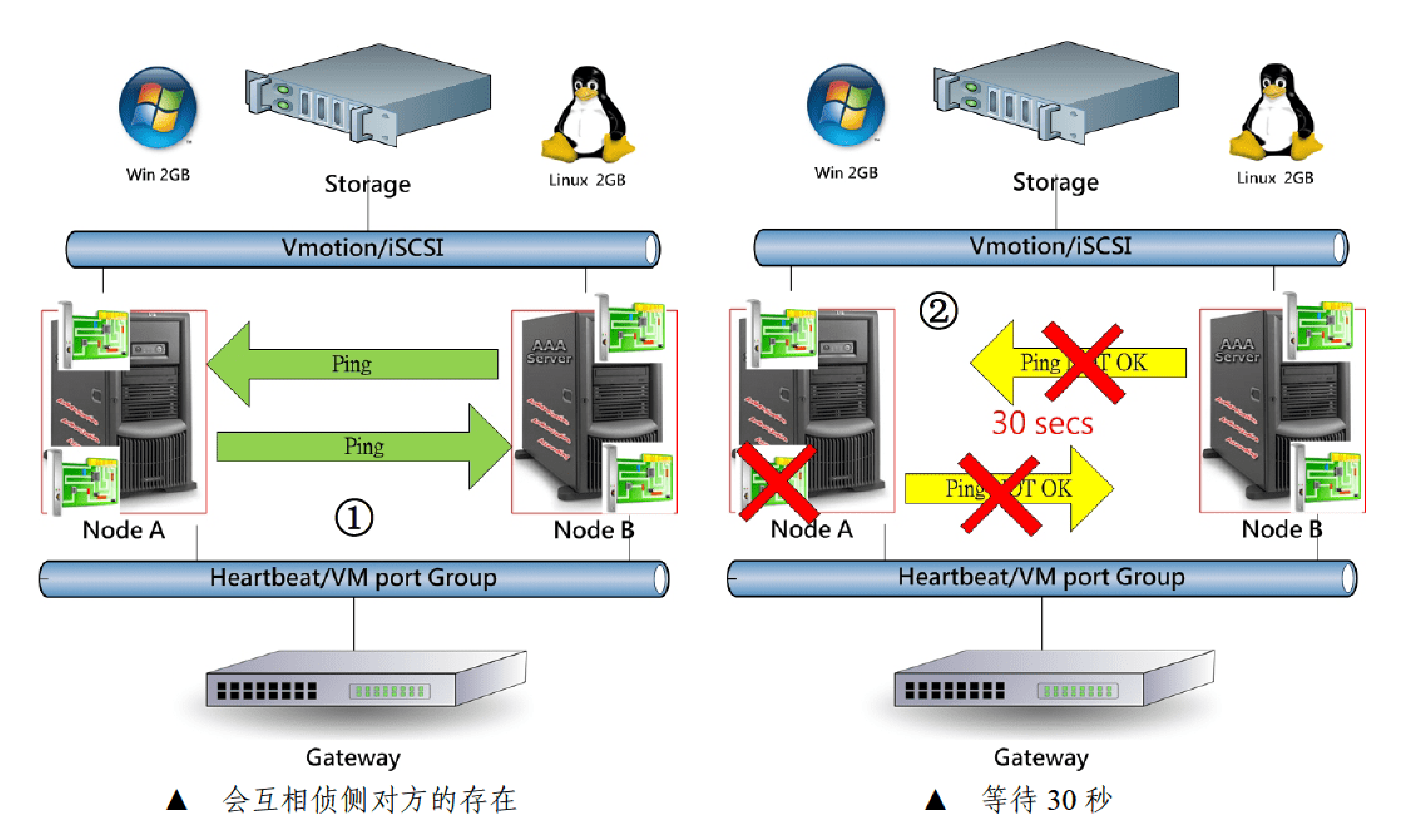
我们在前面的章节不断强调防止单点失误的重要性，而 VMware HA 则是最明显受到影响的例子。如果我们使用的是单个交换机、单个网卡，那么隔离回馈的配置就十分重要了。一般来说，我们如果设计的网络是多交换机、多网卡的，并且将主控台、虚拟机通信端口组和 VMotion 的虚拟交换机都分开用 NIC Team 落实，那么就算心跳线停止了，但虚拟机通信端口组还在，意味着虚拟机还能常规运行，这时候在隔离回馈 IP 无法 ping 通时，我们还就可以让虚拟机不关机，因为虚拟机这时还能常规提供网络服务，只需要将心跳网络快点修复即可。下图所示即为一完整设计的 HA 架构。

7. VMware HA 原理及步骤

在解释了这么多名词之后，我们现在才能正规来落实 VMware HA 原理的说明，在实践 VMware HA 之前，也顺便让读者们复习一下。

8. VMware HA 的原理

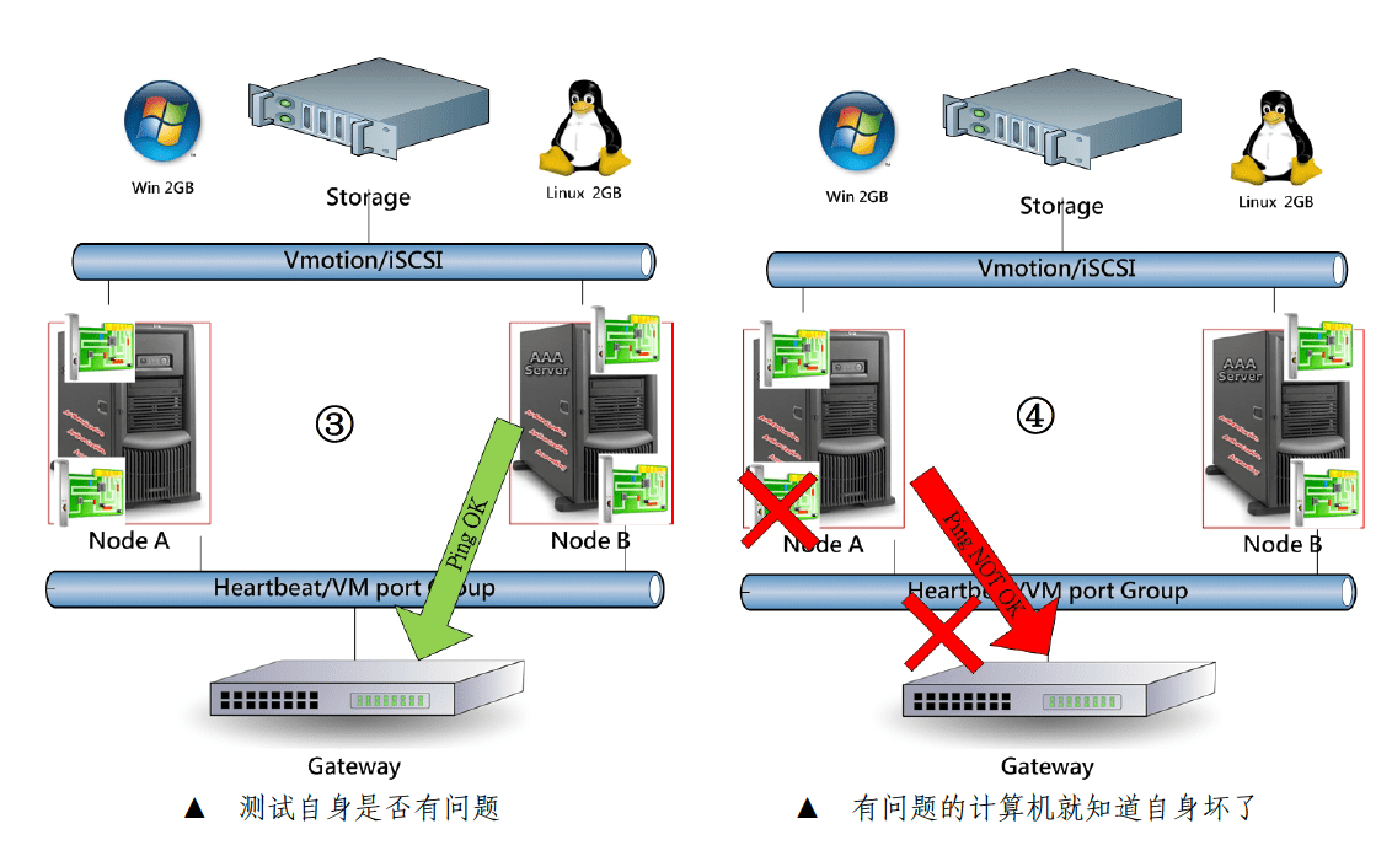
1. HA 集群之间的计算机利用心跳网络互相嗅探其他计算机是否存在，使用主控台内部的 ping 作为心跳网络嗅探的工具。
2. 当发现 ping 心跳网络无法成功时，每一个节点都会等待 30 秒。



▲ 会互相侦测对方的存在　　▲ 等待 30 秒

3. 30 秒之后，发现心跳网络无法创建，就会试着 ping 隔离回馈 IP。如果发现隔离回馈 IP 常规，意味着本身没问题。

4. 如果发现隔离回馈 IP 异常，意味着自身弹出问题，或是自身已经蓝屏，根本无法运行心跳线。

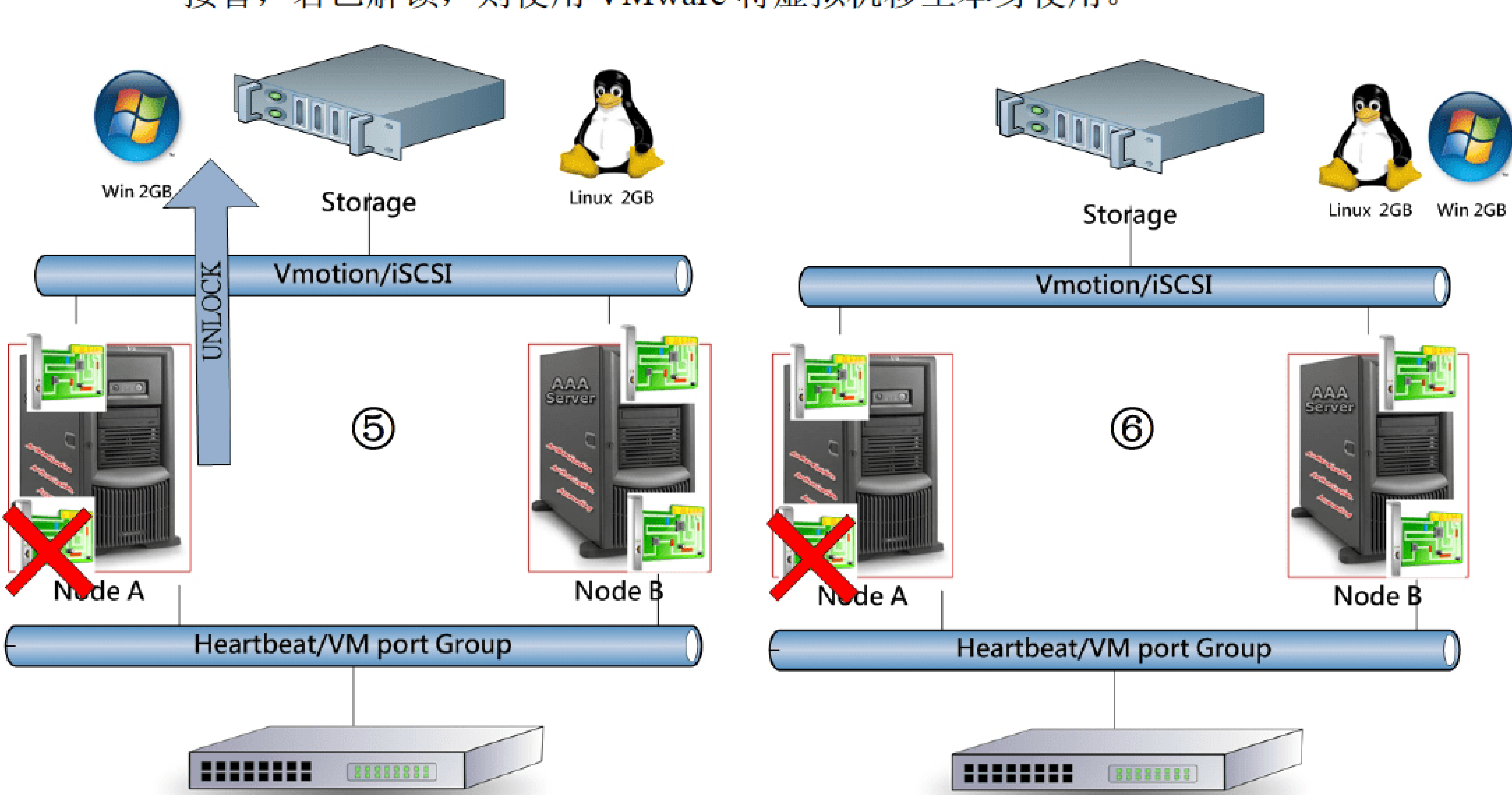


▲ 测试自身是否有问题　　▲ 有问题的计算机就知道自身坏了

5. 若本身无蓝屏，只是心跳线弹出异常，进入存储设备将自身运行的虚拟机解锁，并且放出掌控权。

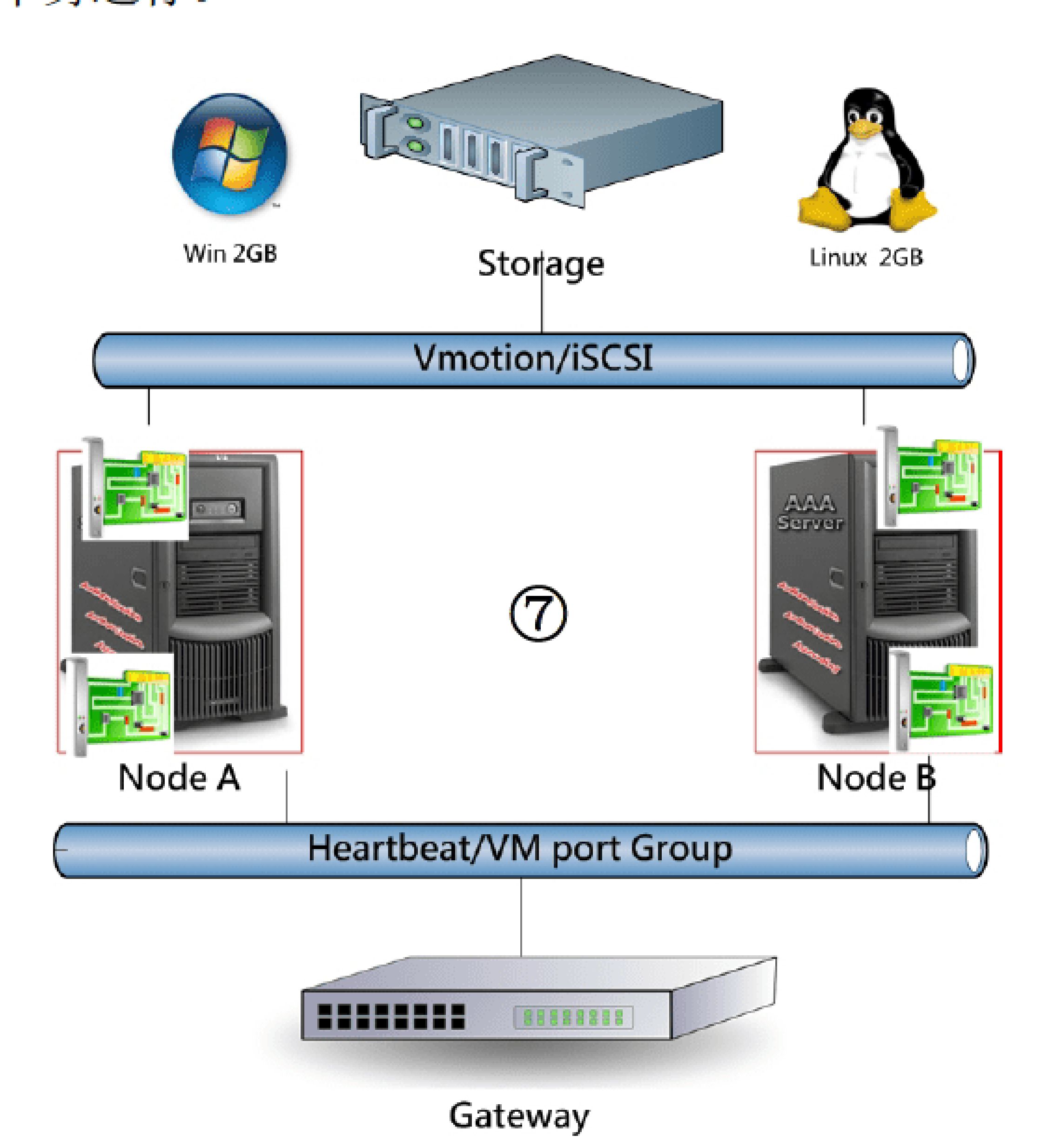
6. 常规节点进入存储设备，并且遍历是虚拟机已解锁。若无解锁意味着虚拟机常规，不

接管，若已解锁，则使用 VMware 将虚拟机移至本身使用。



▲ 进入存储设备将虚拟机解锁　　▲ 备援计算机接管来源计算机上的虚拟机

7．当故障主机心跳线恢复，副本主机将释放原来虚拟机用户，原主机使用 VMotion 将虚拟机移回本身运行。



▲ 回退时，虚拟机会从备援计算机上回到原始计算机上

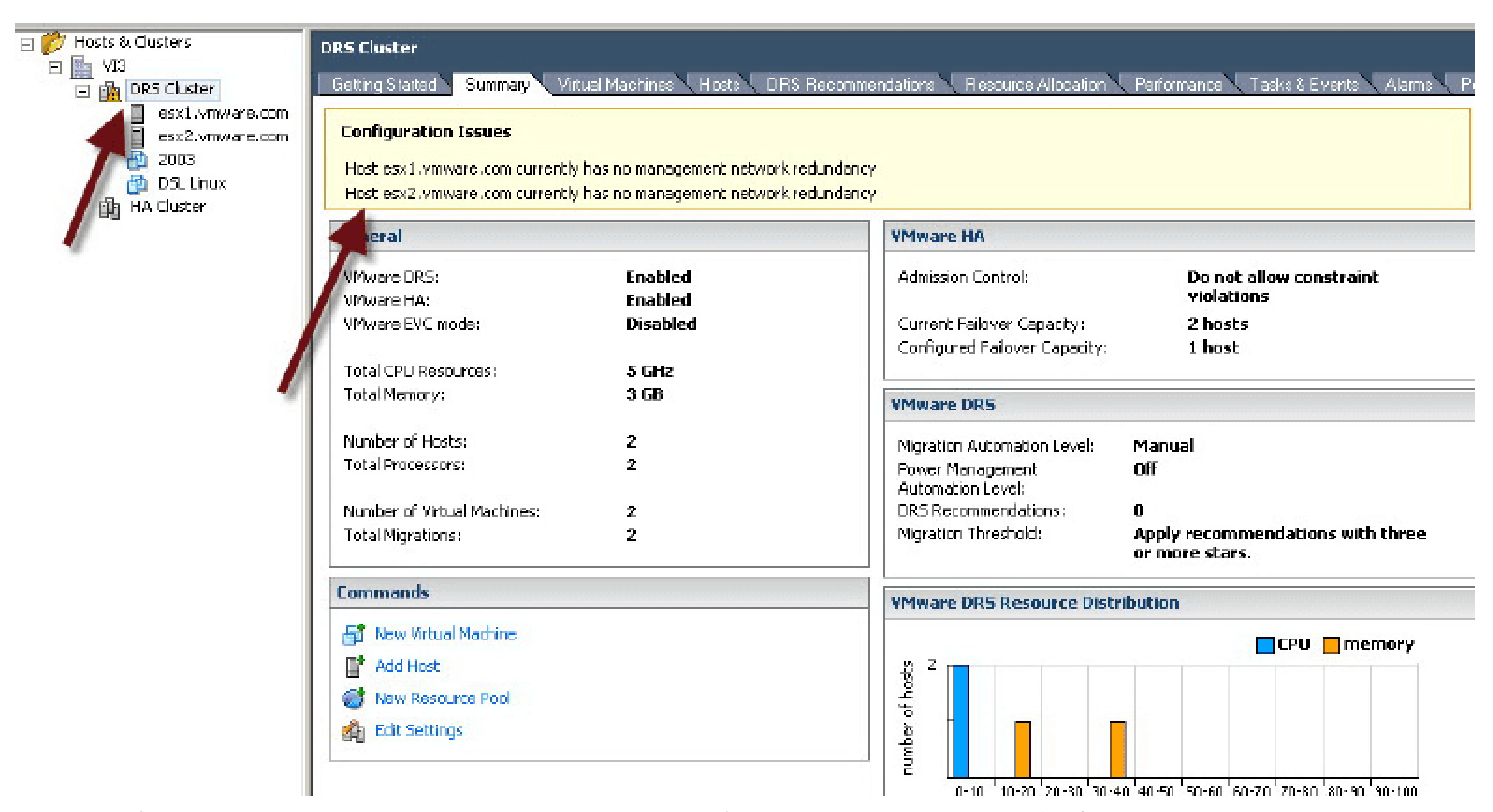
17.2 实践 VMware HA

由于 HA 对网络上的冗余是相当在乎的，而 HA 与 Heartbeat 之间沟通也是经过 Service

Console 来提交心跳数据，因此我们在实践 VMware HA 的时候，最重要的就是先进行心跳线的冗余。之后的实践就是配置 VMware HA 的参数问题了。在这一小节，我们就来看看 VMwarevSphere 架构的最后一块拼图 HA 实践详细过程。

17.2.1　使用 NIC Team 来落实“心跳线”的冗余

在上一小节我们看到了使用两片网卡来落实“心跳”网络的冗余，事实上就是使用两个 Service Console 主控台来落实冗余。但两个主控台还是无法防止单网卡的失误，因此我们需要让一个虚拟机路由器使用两片以上的网卡，此时就是使用 NIC Team 的时机了。

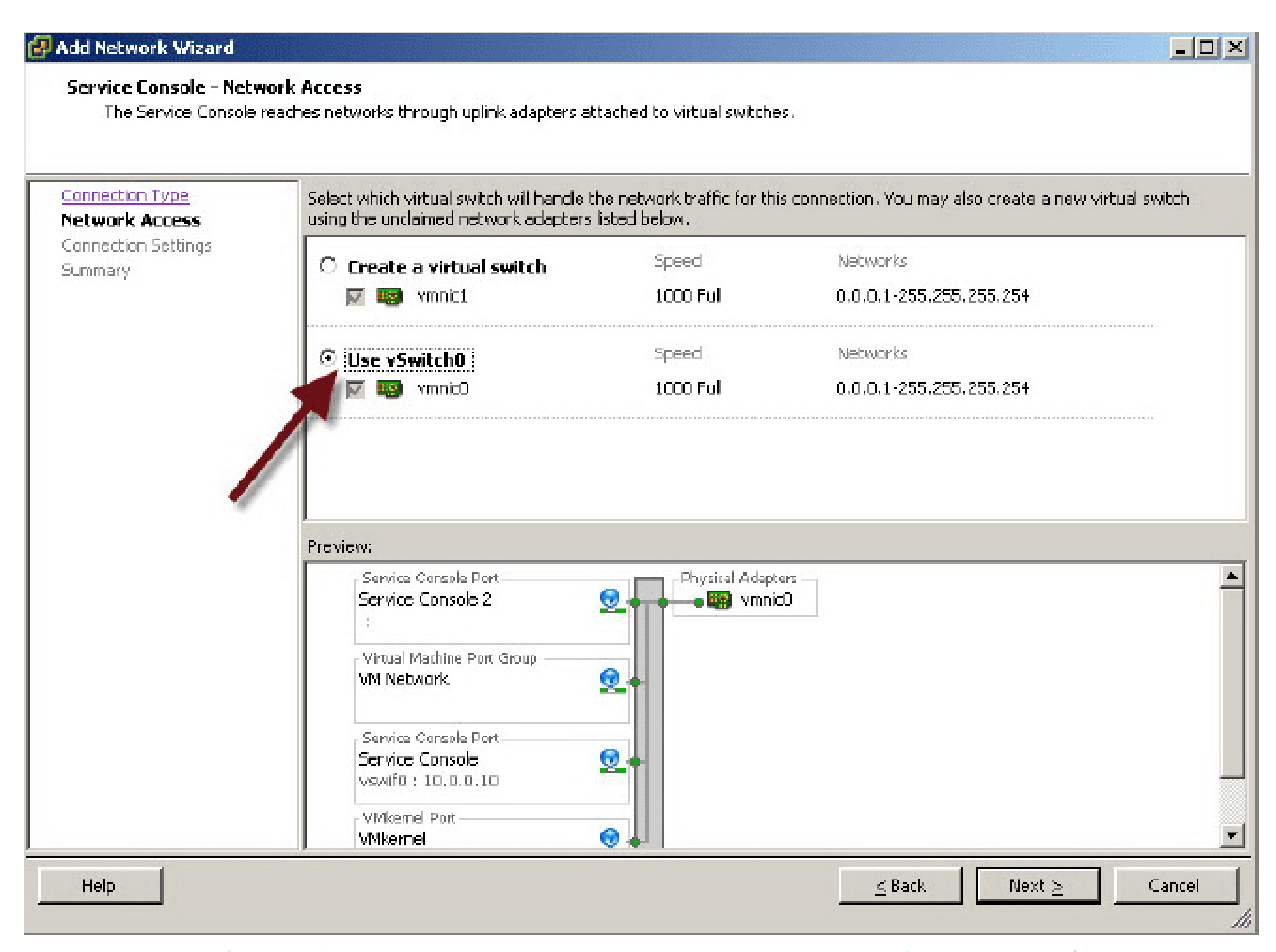


▲　没有创建“心跳线”（主控台）冗余时，会有错误消息，但还是可以落实 HA

在激活 NIC Team 之前，首先要确定你的系统有多出来的网卡，并且联机到另一个物理交换机上，接下来就是 NIC Team 在 HA 上的实践。

1. 创建新的主控台

首先我们到 esx1/esx2 上创建第二个主控台。在创建的时候别忘了选择现在主控台所在的虚拟交换机。至于创建主控台的步骤，对读者来说应该已经相当熟悉，我们就不再详述了。



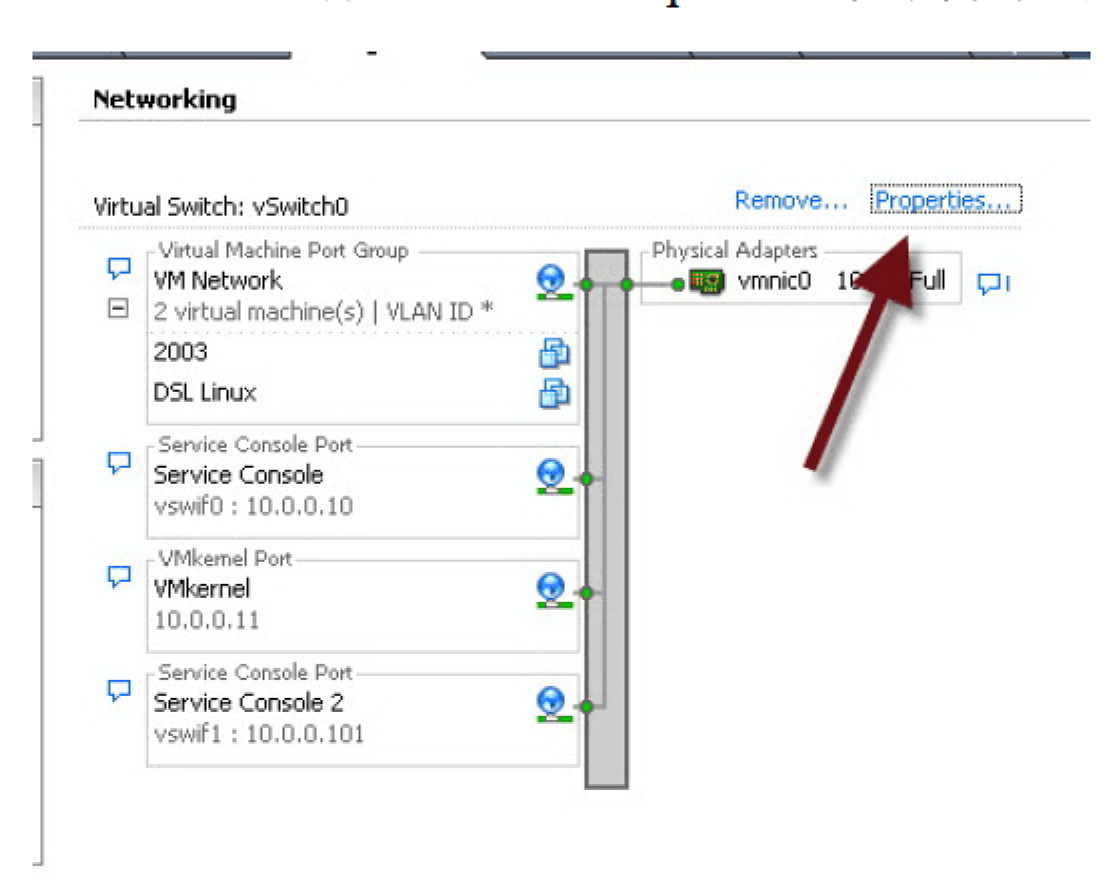
▲　创建备援的主控台时，要在现有的虚拟交换机上

2. 创建 NIC Team

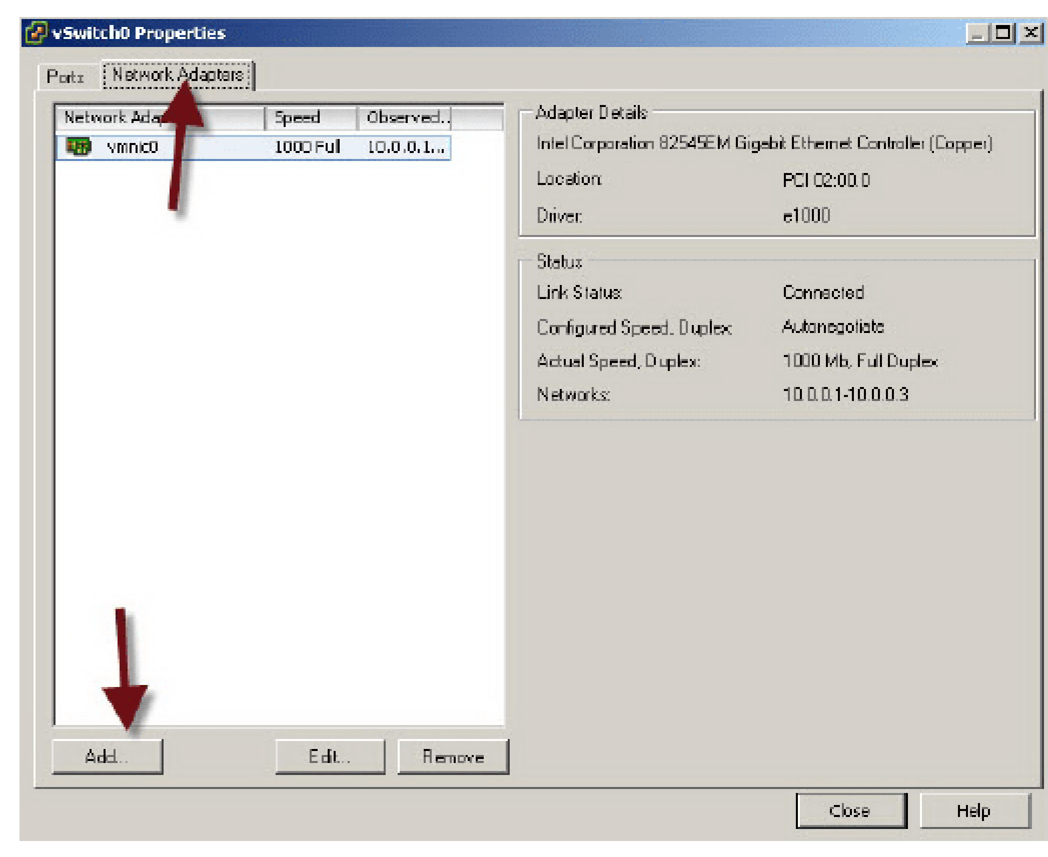
NIC Team 顾名思义，就是让多片网卡落实冗余的任务。在本例中，我们希望有两个物理网卡连上两个不同的物理交换机，来做到 vSwitch0 上主控台的冗余。接下来就是进行 NIC Team 的详细步骤。

▶ NIC Team 的详细步骤

1. 进入 ESX 主机的网络配置部分选择 Properties。
2. 选择 Network Adapters 选项卡并单击 Add 按钮。

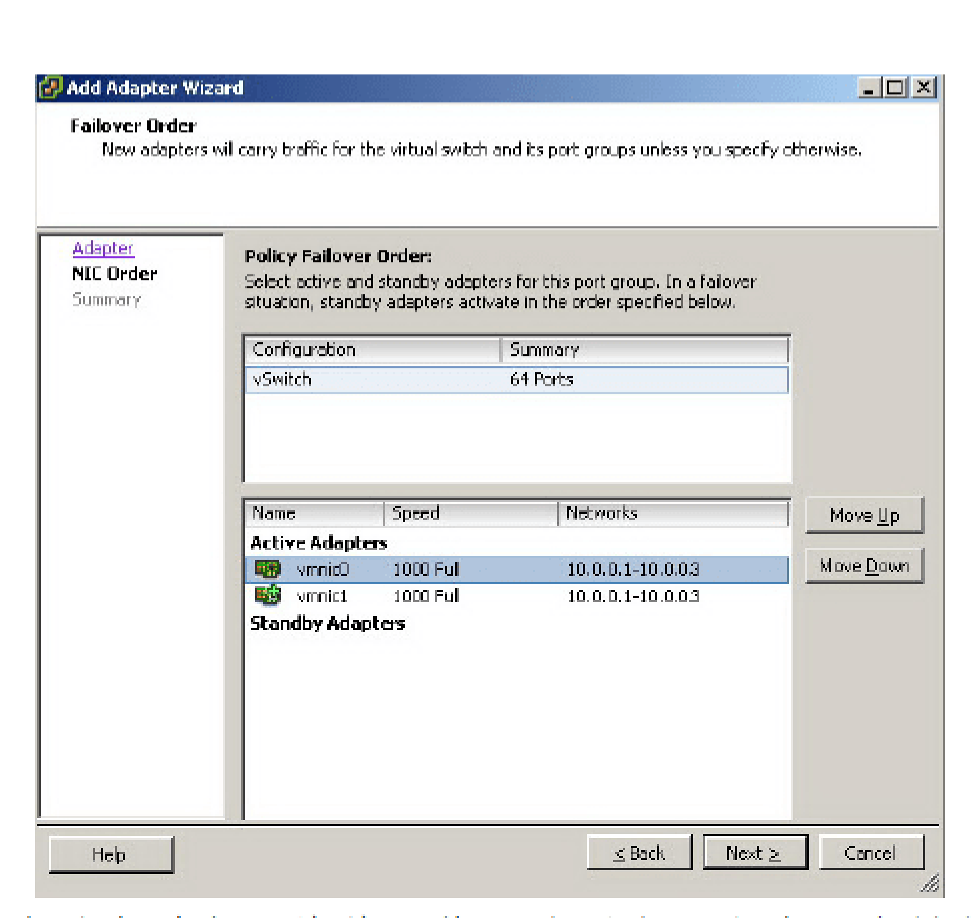


▲ 选择属性的部分

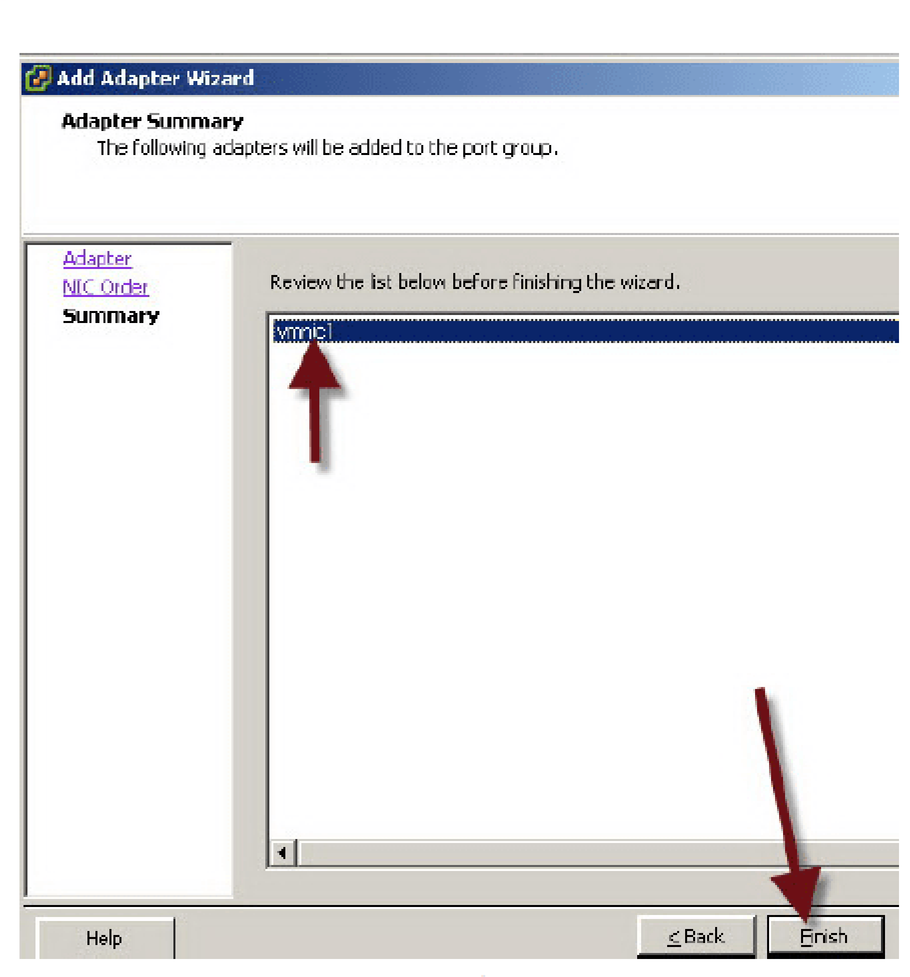


▲ 选择新建网卡

3. 将 vmnic 选中，并且单击 Next 按钮继续。
4. 此时你会看到第一片 vmnic0 和 vmnic1 都成为同一个虚拟交换机的网卡，并且并发作用。单击 Next 按钮继续。
5. 在弹出 vmnic1 之后单击 Next 按钮继续。

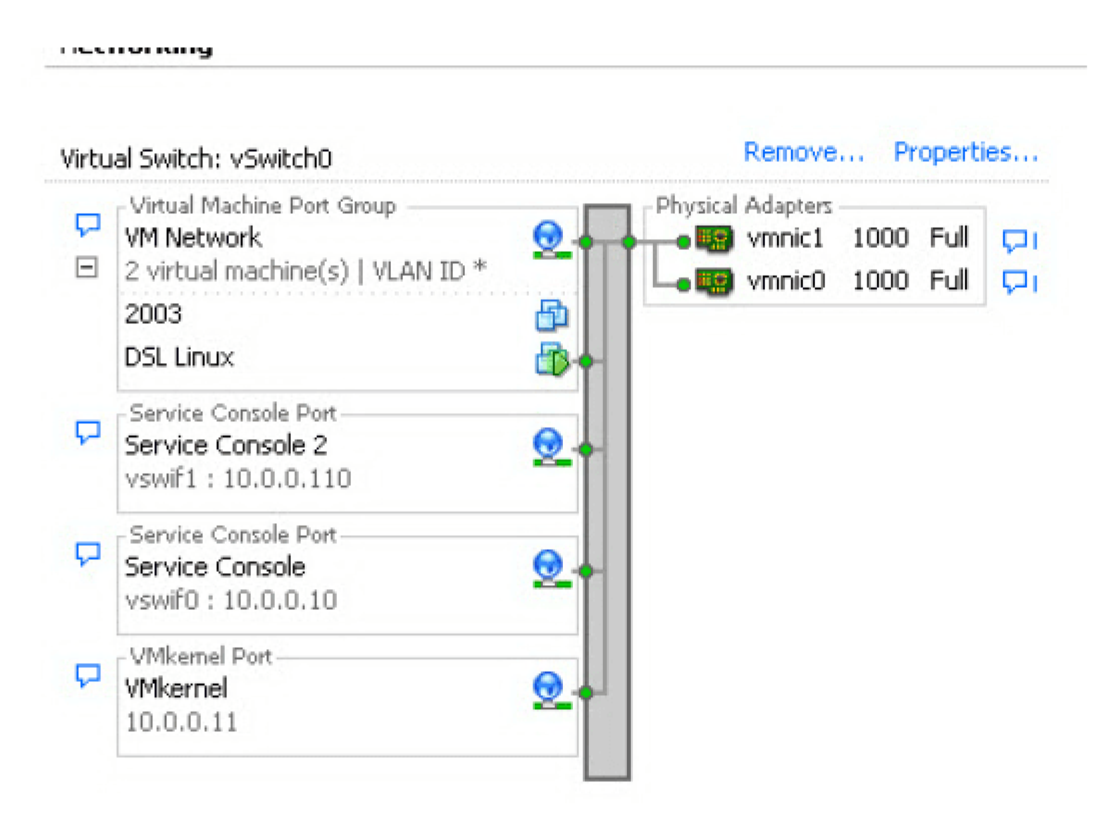


▲ 此时会弹出两片共同作用的网卡，没有一个是待命的



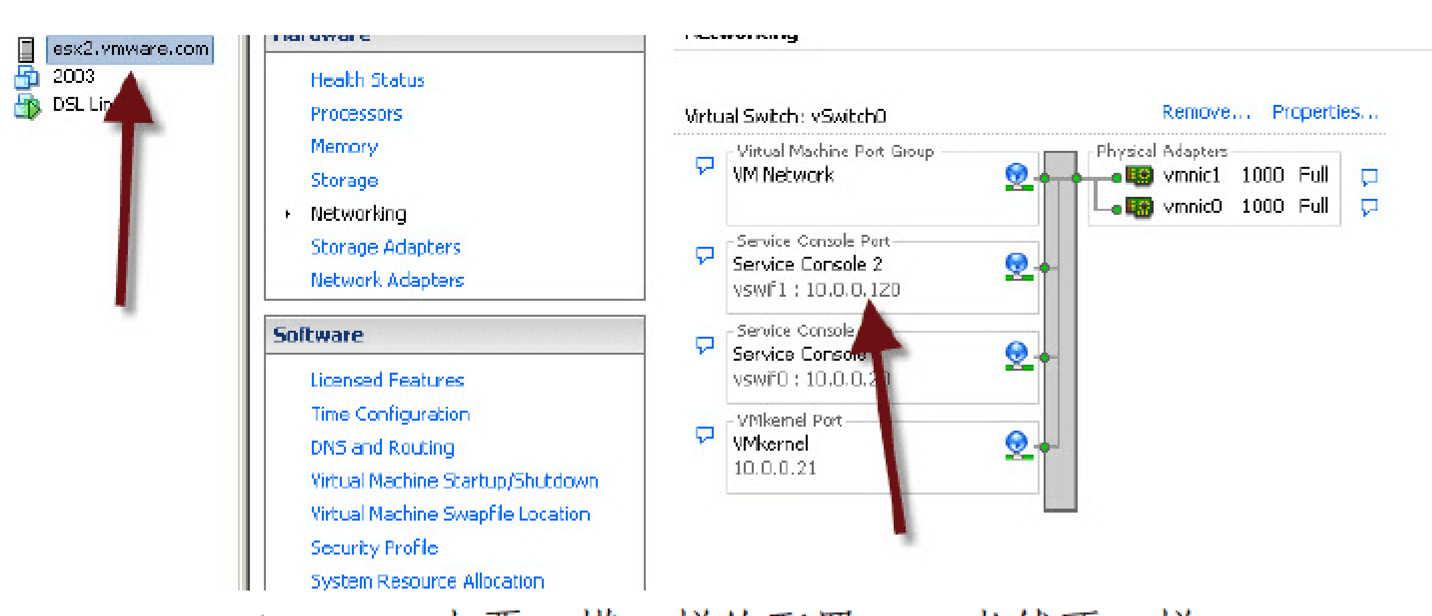
▲ 再度回车

6. 当弹出如下图所示的画面时，意味着你的网络已经成功地配置了 NIC Team，而报警标志也不会再弹出。



▲ 设置成功了

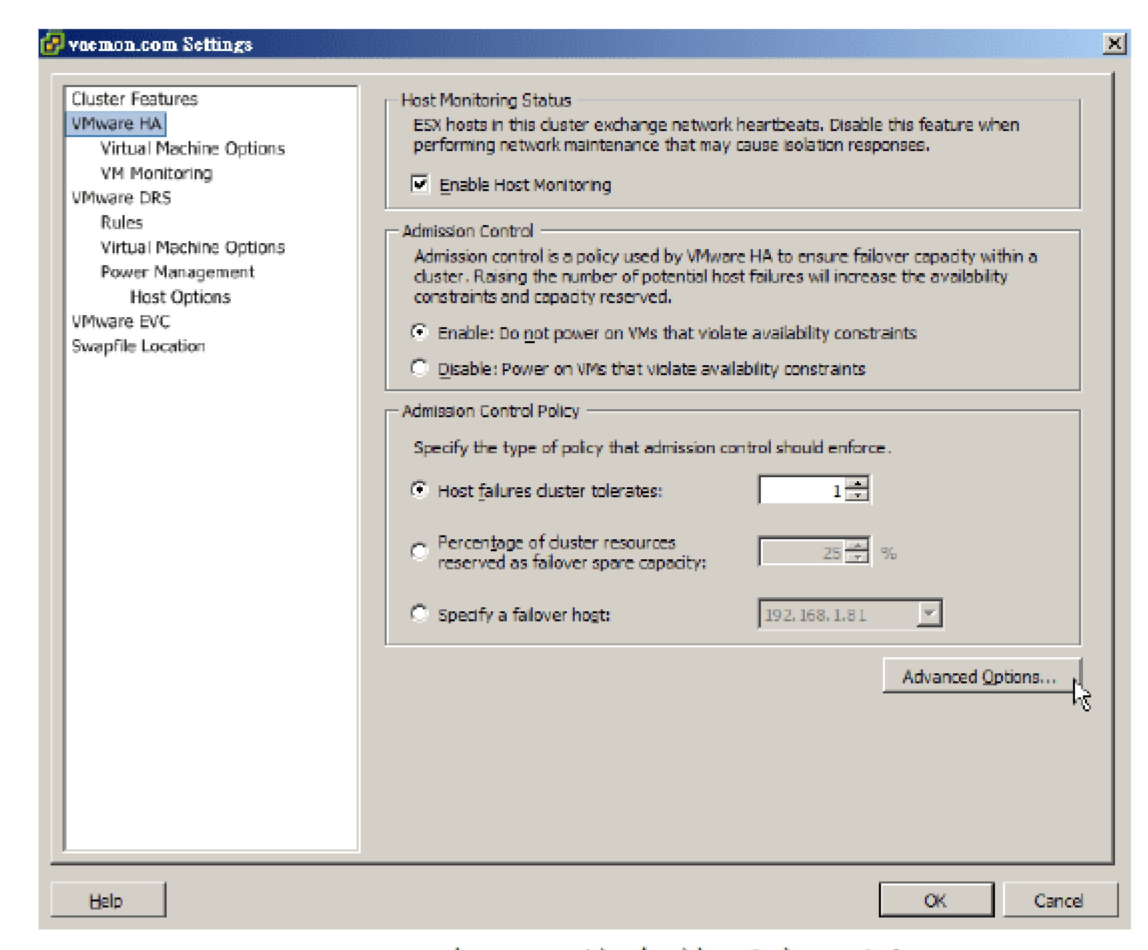
7．别忘了到 esx2 也进行相同的步骤。



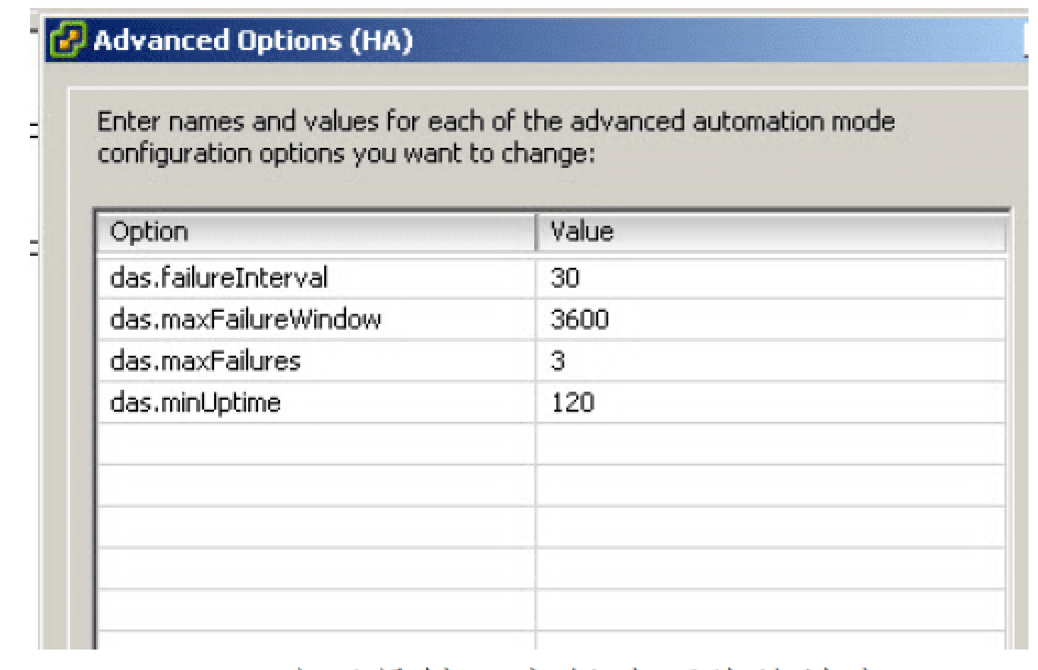
▲ esx2 也要一模一样的配置，IP 当然不一样

3．常用的 HA 参数简介

我们前面提到，当心跳线 30 秒之后就会开始进行隔离回馈 IP 的测试，这些“30”秒，隔离回馈的 IP 的数值，当然也能在 VMware 的 HA 中设置。配置的手段很简单，也是在集群的 HA 选项中。只要选择 VMware HA 选项，单击 Advanced Options 按钮即可。我们就来看看其中常用的数值。



▲ 在 HA 的参数项中配置



▲ 这里是键入高级选项值的地方

4. VMware HA 常用选项配置

- das.failureInterval：HA 激活时间，如果心跳线在这个时间内没有回馈，就激活 HA，默认值为 30 秒。
- das.maxFailureWindow：重设主机的时长，就是说在这个时间内会重新测试主机是否成功，单位为秒，默认值为 3600 秒（即一小时）。
- das.maxFailures：在上述时间内重设主机的次数，默认值为 3，即在一小时内会重试主机 3 次。
- das.minUptime：HA 激活前，虚拟机必须常规运行的时间数，单位为秒。也就是说，如果虚拟机常规使用的时间少于 120 秒，HA 功能就会准备发动其他的节点接管这台虚拟机。
- das.isolationaddress：隔离回馈的 IP，如果不配置的话，就会以主控台的 Gateway 为默认值。
- das.isolationaddress1/2：可以配置第二或第三个隔离回馈 IP。
- das.defaultfailoverhost：当主计算机发生故障时，第一个接手的副节点，使用 FQDN 名称。
- das.failuredetectiontime：嗅探故障的时间点，默认为 15 秒。

17.2.2 使用 VMware HA 功能

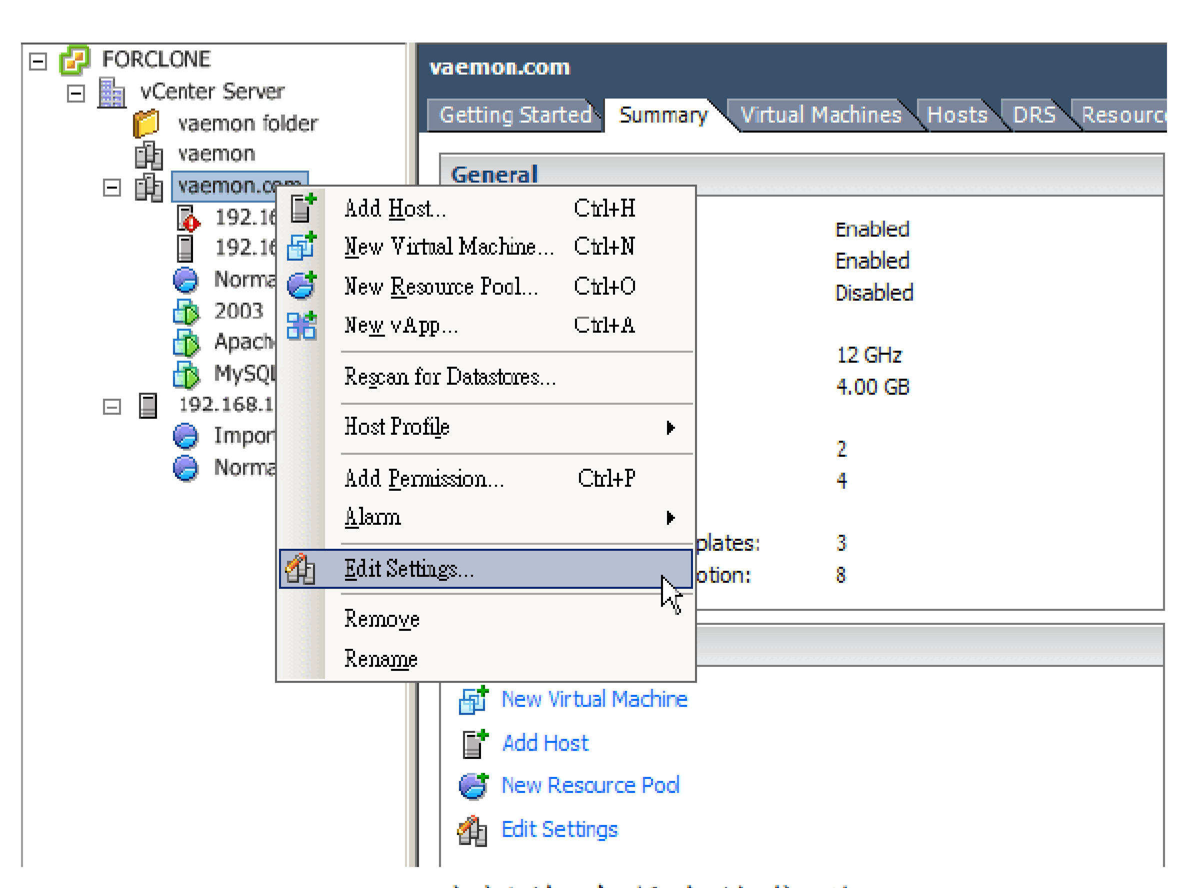
VMware HA 的使用相当简单，反而是原理说明和事前的规划较为复杂。当然在我们理解了 HA 的原理以及将所有的前置任务落实后，就可以在 vCenter Server 中创建 HA 的集群了。

1. 创建 VMware HA 集群

创建 HA 集群很简单，只要在创建集群时选择 HA 选项即可，你也可以将现有的集群加入 HA 功能，而在加入之后，原有的主机也必须配置 HA 的参数。我们就以现存的集群来加入 HA 功能作为示例。

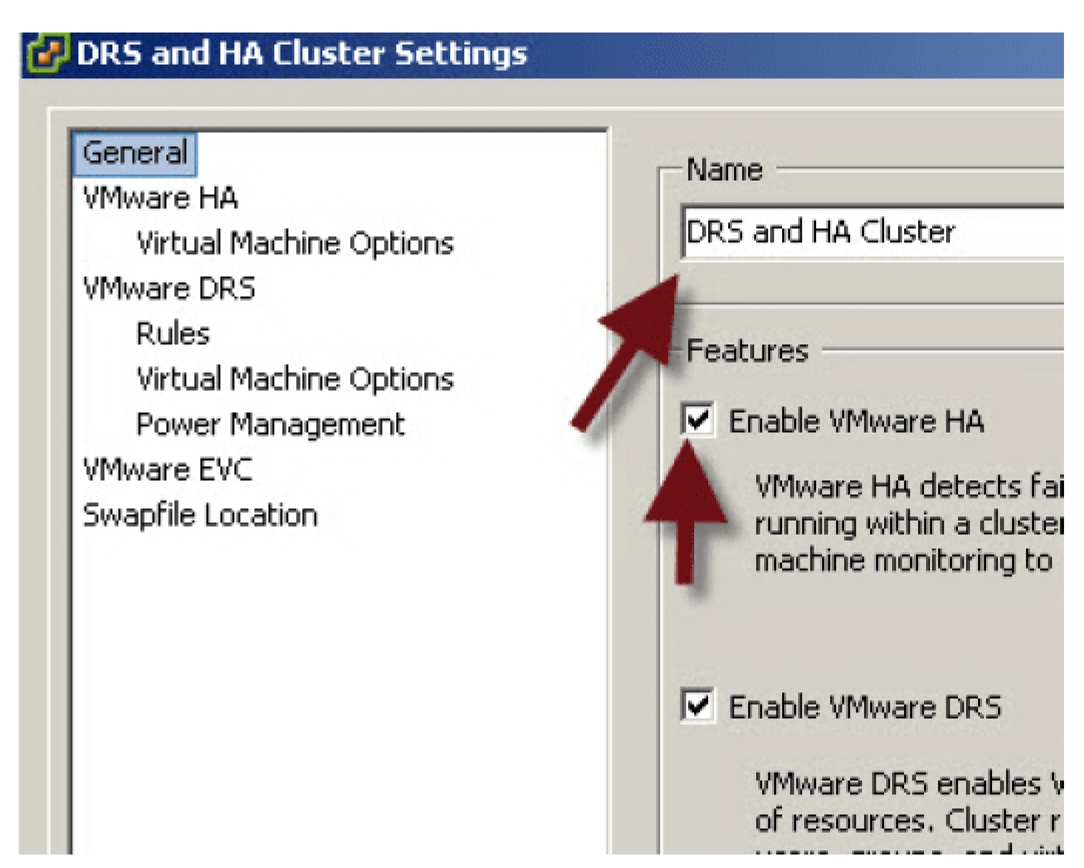
2. 将集群的 HA 功能打开

1．我们以上一章使用的 DRS 集群为例，在该集群上右击并选择 Edit Settings 选项。



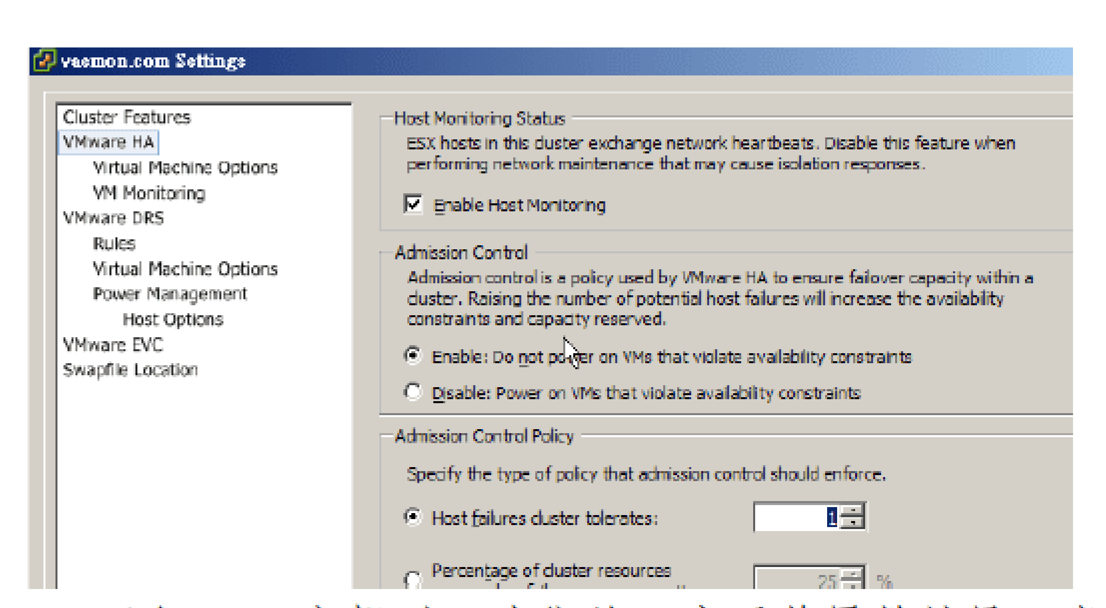
▲ 选择修改现有的集群

2．在 General 的选项卡下，将 Enable VMware HA 选中。也将集群名称换成 DRS and HA Cluster。



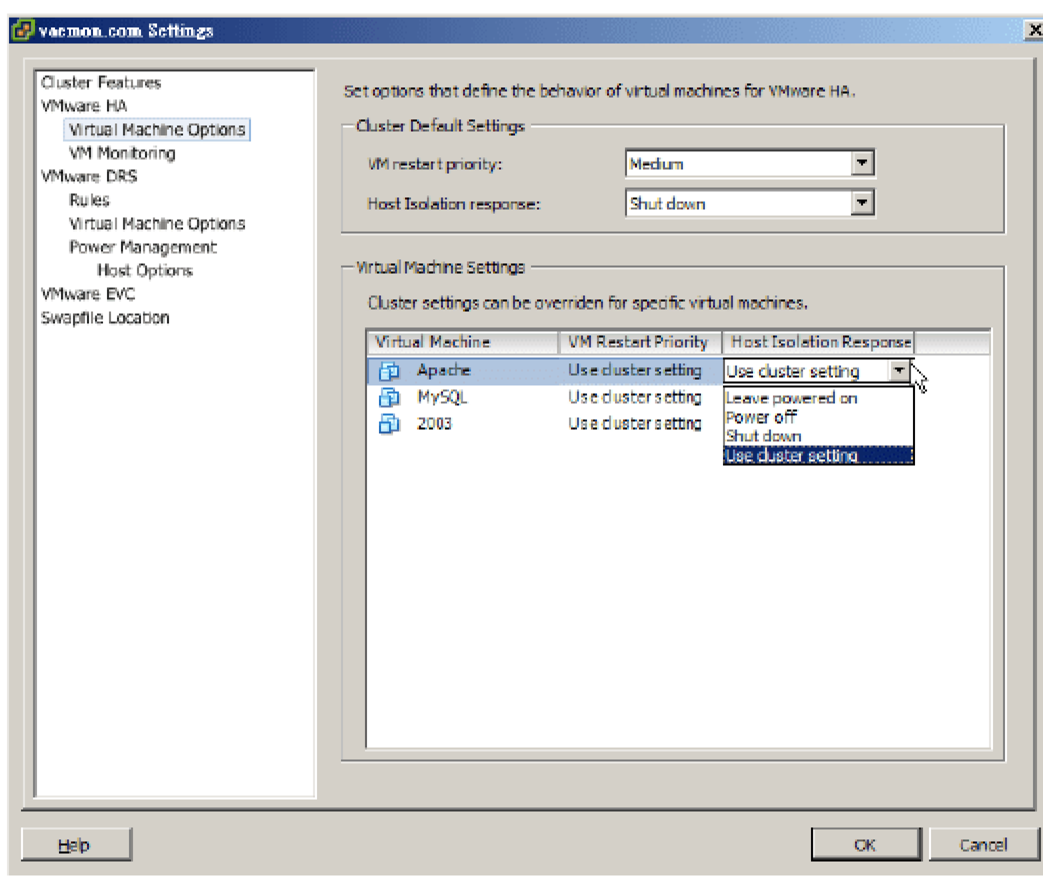
▲　将 HA 的选项勾起来

3．选择 VMware HA 的选项卡，并且将可容忍蓝屏主机设为 1 台，让虚拟机能在资源不够时还是重新激活。其他的配置不变。



▲　两台 ESX 主机时，我们的可容忍蓝屏数就是一半

4．选择 Virtual Machine Options，并且在任何一台虚拟机上单击其重新激活参数和 Host Isolation Response 参数。由于我们已经配置了 NIC Team，因此可以让 Isolation Response 的虚拟机不要关机，单击 OK 按钮即可。



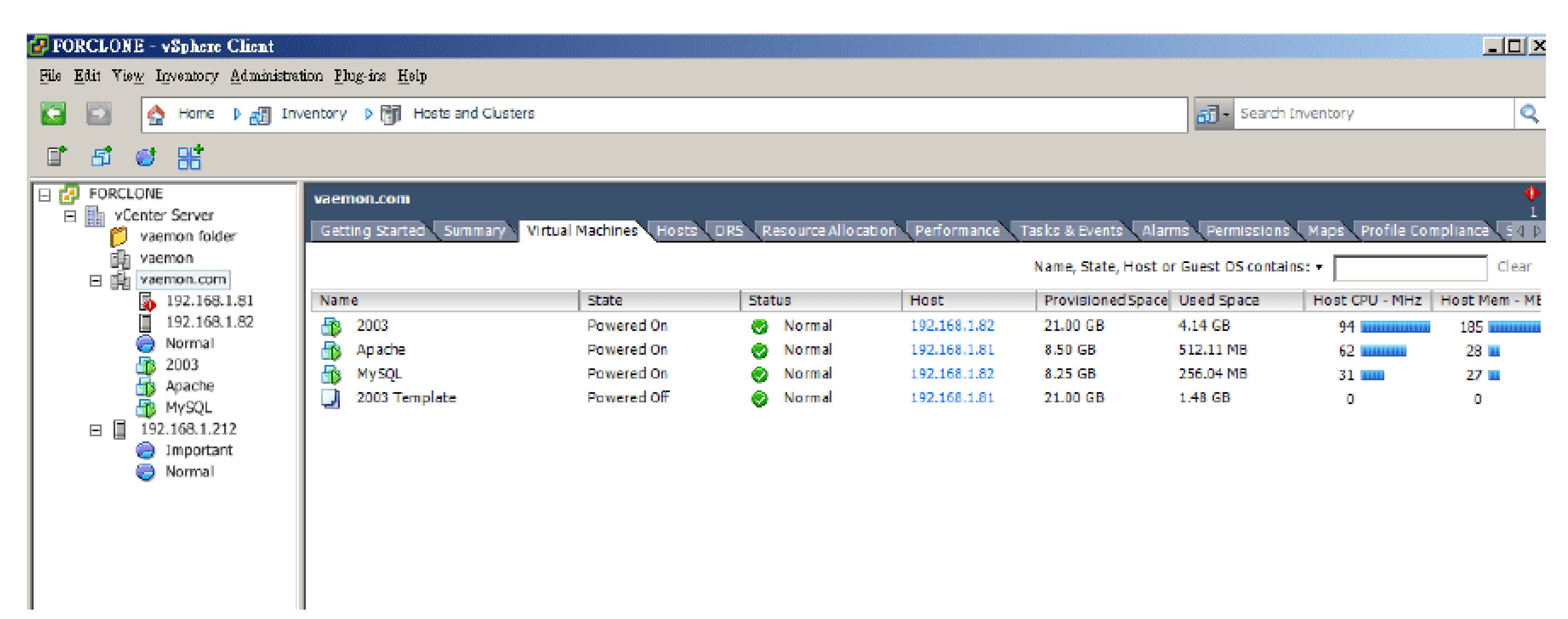
▲　我们可以针对虚拟机本身再做配置

3. 测试 VMware HA

终于到了这兴奋的时刻，可以测试 HA 功能了。在 VMware 的 HA 中，事实上蓝屏时间视你的网络设备和虚拟机的操作系统种类、内存大小、运行的服务而定。我们就以一个最简单的双节点 HA 为例子，来看看 HA 的动作。

4. 测试 VMware HA 功能

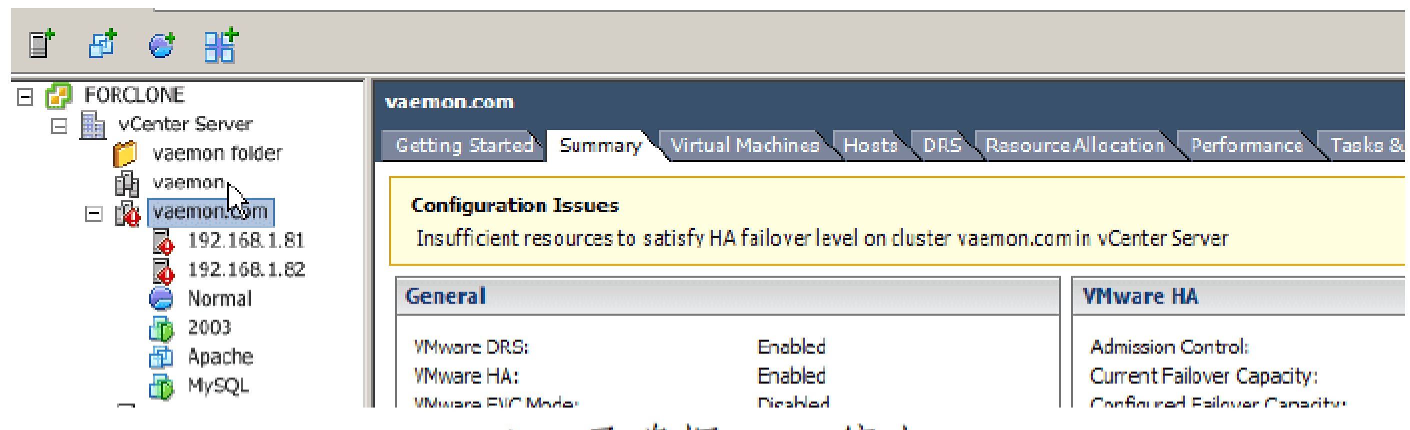
1．首先进入 vCenter Server，并且将虚拟机 2003 打开在 esx1 上，将虚拟机 DSL Linux 打开在 esx2 上。



▲ 在两台 ESX 主机上各开一个虚拟机

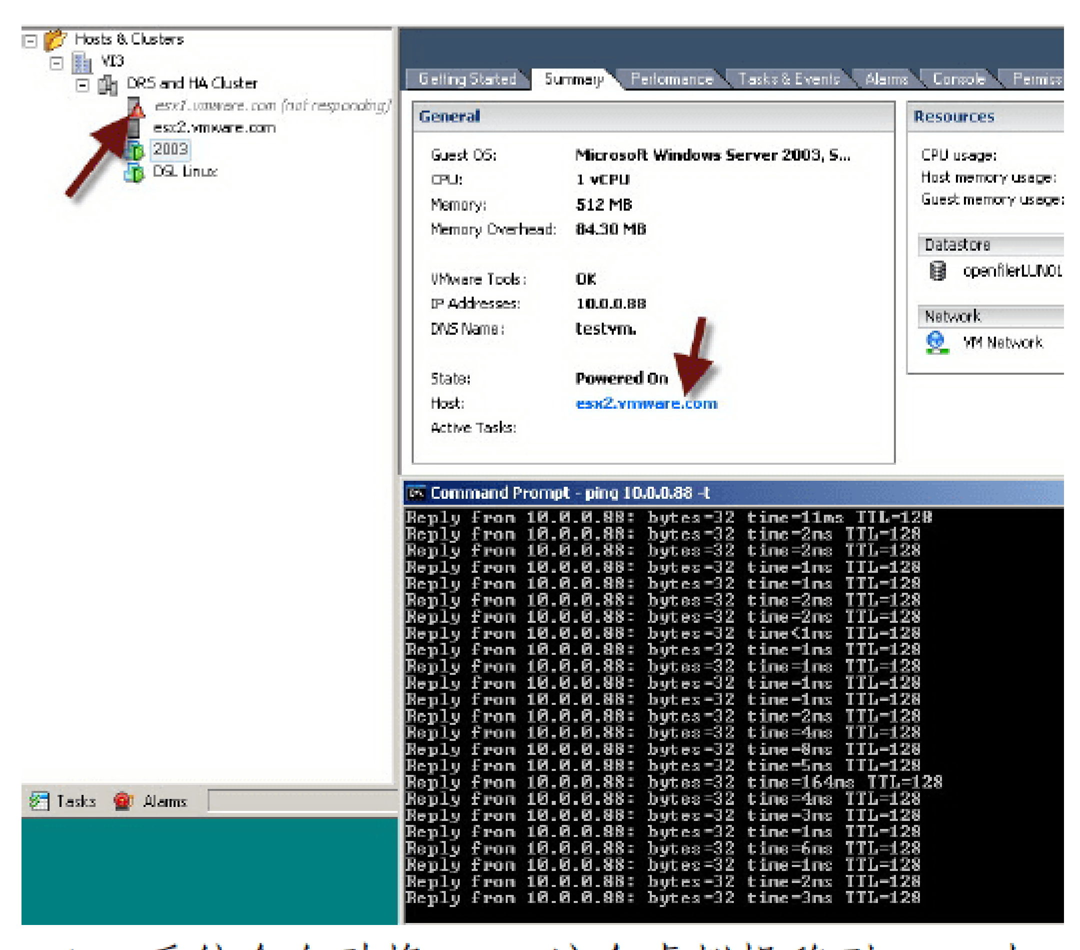
2．此时我们将 esx1 上的两个网络线都拔除，让 esx1 的心跳线完全停止。

3．经过 30 秒后，系统侦测到 esx1 已经心跳停止。



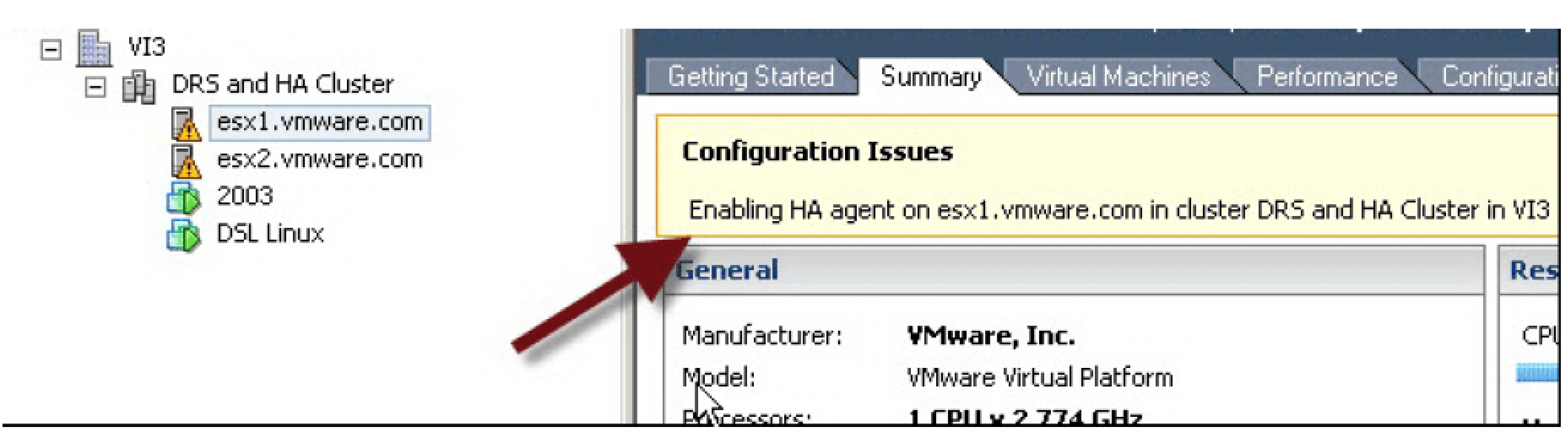
▲ 已嗅探 esx1 停止

4．经过一阵子，你会发现原来在 esx1 上运行的 2003 已经跑到 esx2 上了。



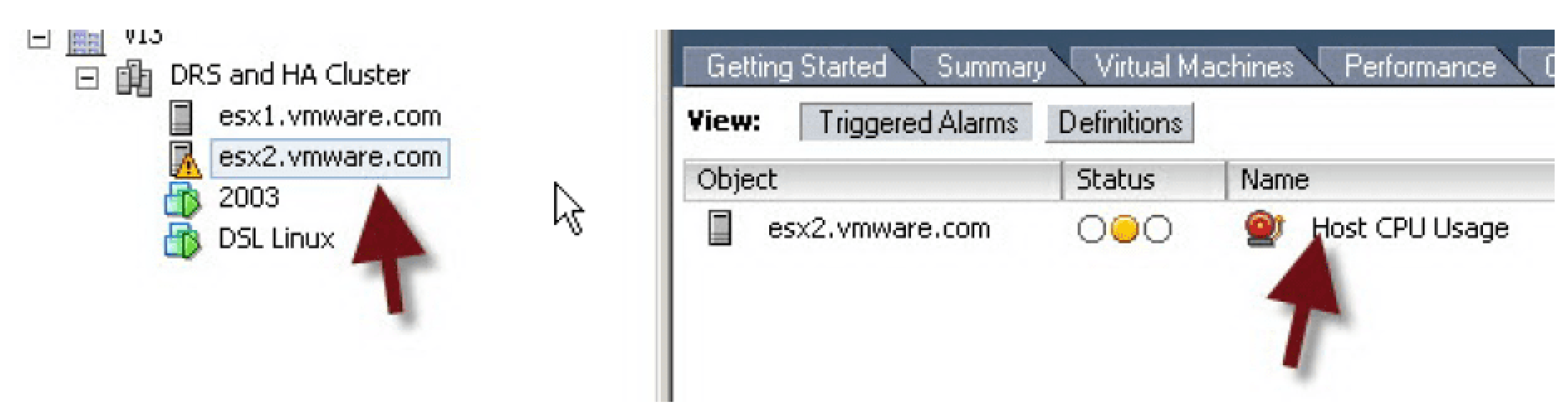
▲ 系统会自动将 2003 这个虚拟机移到 esx2 上

5．当我们将 esx1 的网线再插上去，会发现 esx1 又重新加入 HA 集群中。



▲ esx1 在恢复时会重新加入 HA 集群，加入时会有报警标志

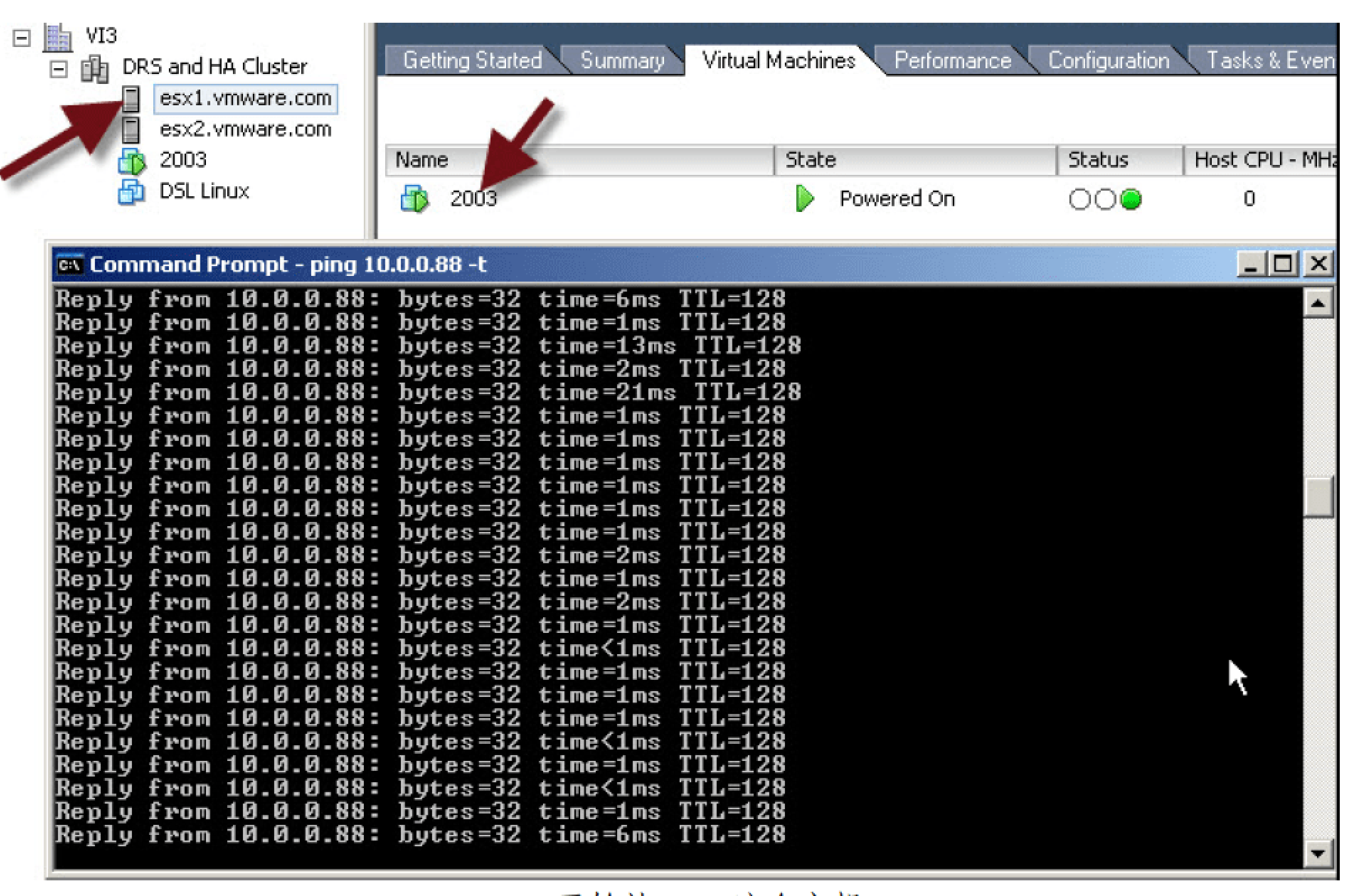
6．最后 2003 还是移回 esx1，而 HA 的动作也停止。



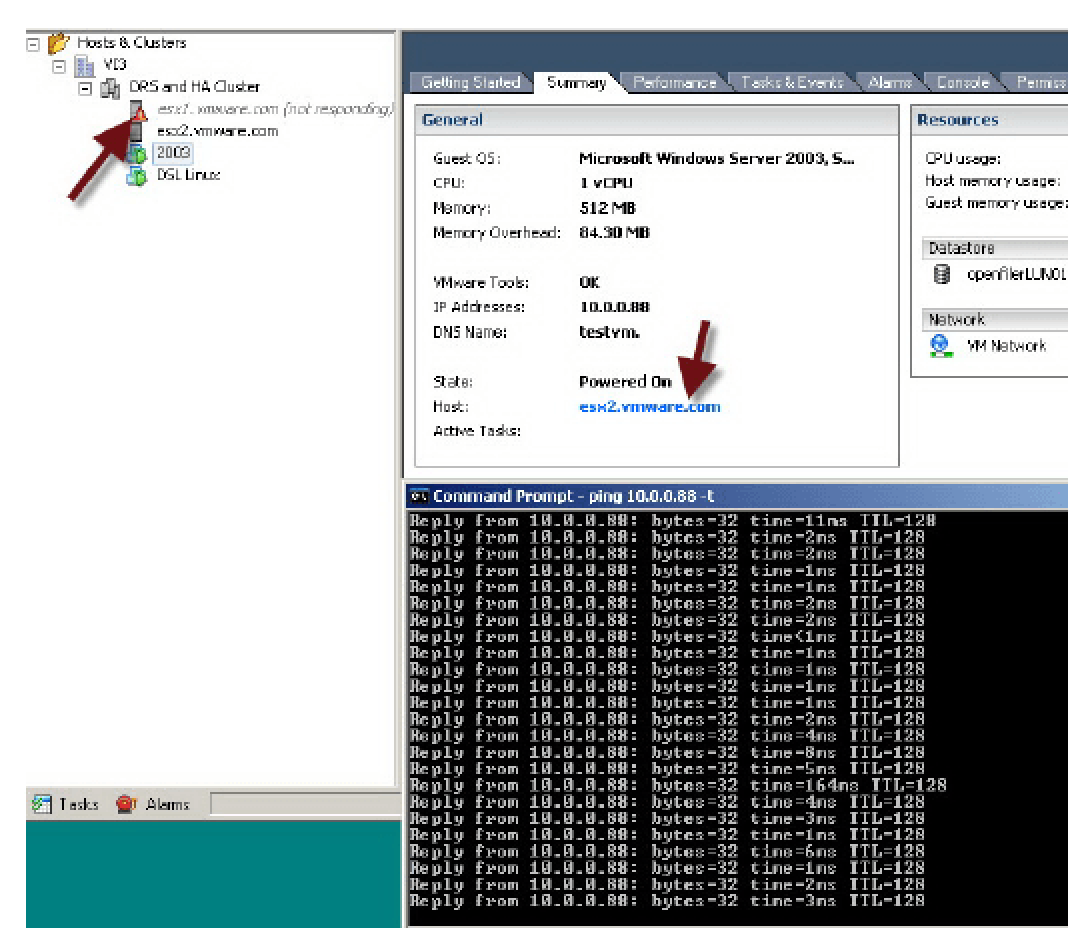
▲ esx2 因为两个虚拟机资源吃紧，但好险 esx1 已经回退了

5. 近距离观察 HA 动作

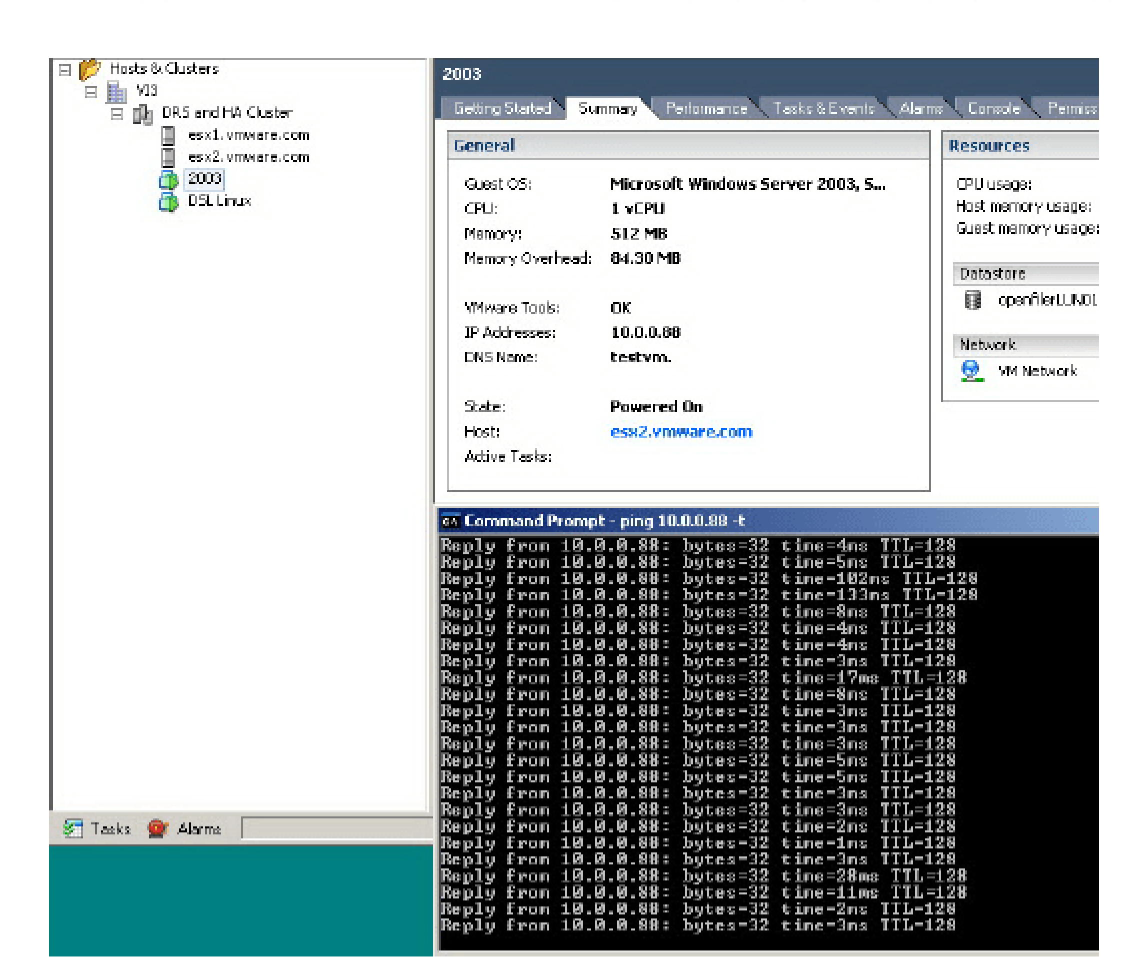
在使用 HA 时，一定会有一小段时间会发现系统是无法运作的，这是因为系统必须移到另一台好的计算机上重引导。但 VMware 的 HA 功能的确定能有效解决 ESX 主机蓝屏时其上虚拟机的处理状况，让整个 vSphere 架构充分发挥永远不蓝屏的功能。



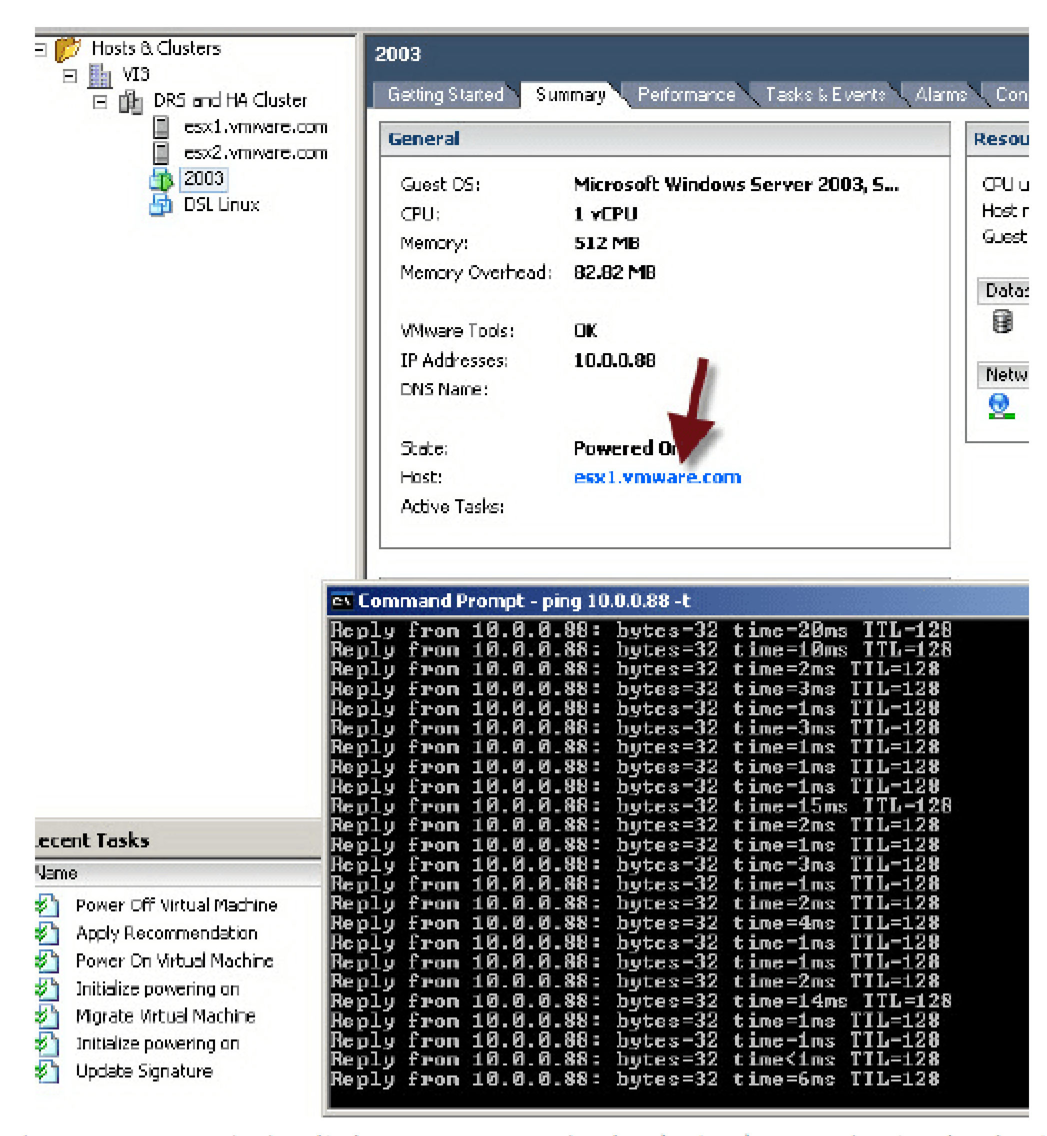
▲ 一开始就 ping 这台主机



▲ 就算 esx1 停止时，其上的虚拟机仍然没有报文漏失



▲ 在移到 esx2 上时也常规



▲ 移回 esx1 时也常规，2003 这个虚拟机一直没有停止服务

结 语

从第 15 章开始的 VMotion 准备任务，一直到本章的 HA 配置，相信读者一定能体会到 vSphere 架构的强大威力了！和微软的 Live Migration 比起来，vSphere 架构更加完整，并且提供更多的选项和细心的考虑，真正创建了一个能充分利用资源、合理分配 CPU/内存的永不蓝屏环境。再配合了 DRS 之后，所有的系统会自动查找最合理的主机，让用户和系统管理员永远不需要担心服务的中断和性能。

读书笔记

第 18 章 创建永不停止的服务架构：Fault Tolerance

关键词：

- 理解什么是双机热备份
- VMware FT 解决的问题
- VMware FT 和 HA 的异同
- 实施 VMware FT 的前提
- 支持 FT 的 CPU 列表
- VMware vSphere 的先决条件

企业环境最担心的就是服务中断，能让服务中断（Downtime）的时间减至最少是所有企业绞尽脑汁想要达成的目的。在没有 vSphere 的时候，大部分的技术顶多将停机时间减低，却无法完全消除服务中断，但有了 vSphere Fault Tolerance 技术之后，在一个设计良好的 vSphere 环境中，竟然可以让服务中断的时间趋近于 0，这对 IT 人员来说，简直是做梦都不敢想象的事，但是在 vSphere 中，却只是几个鼠标的操作而已，我们就来看看大家期待已久的 FT 功能。

18.1 从双机热备份原理开始

早在数十年前，大型机的用户就很清楚单台计算机用在重要的任务上是十分危险的。从掉电到硬件、操作系统、应用软件的故障，将会使系统整个蓝屏，因而导致服务中断。因此使用多台计算机来提供单个服务，当某台计算机故障时，其他的计算机能在最短的时间内接手服务。因此让服务中断的时间能减到最少，一直是系统管理员的巨大挑战。



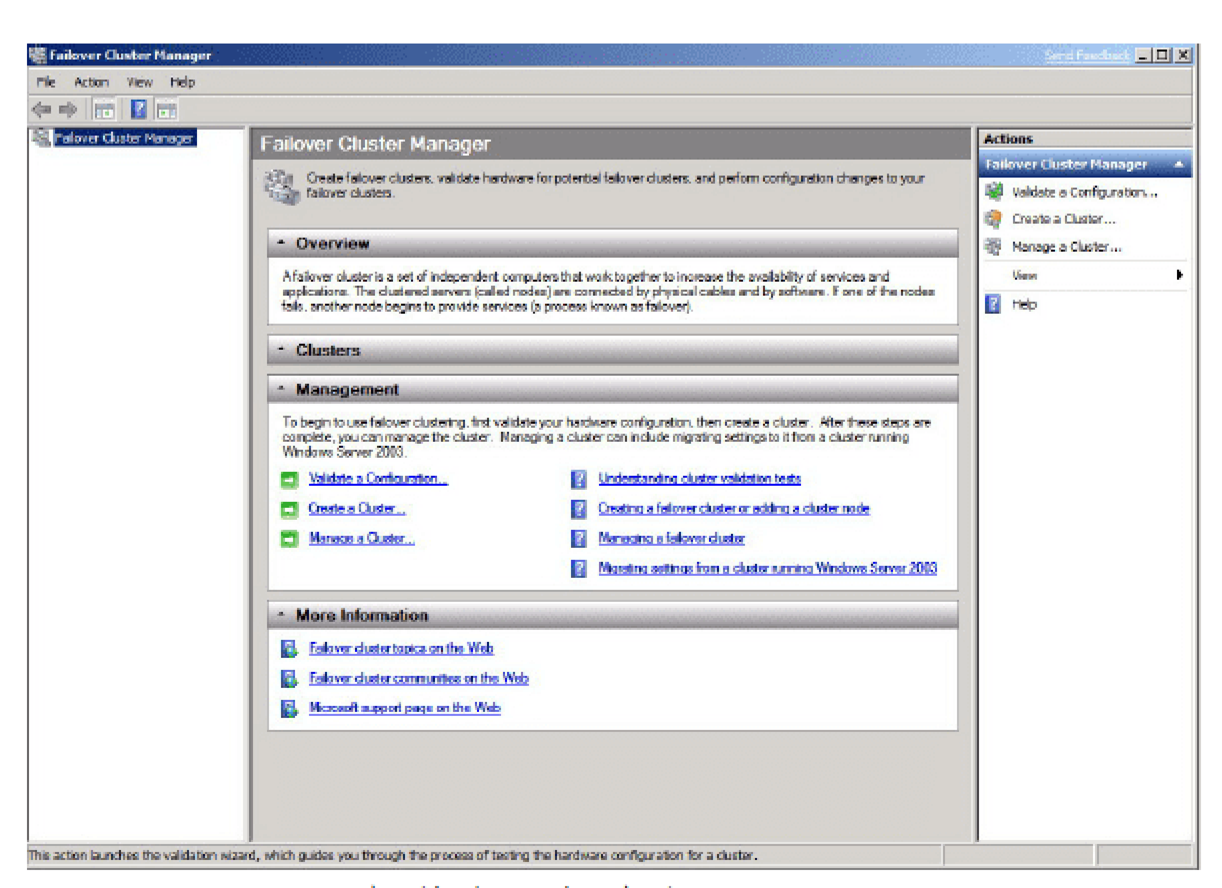
▲ 企业环境使用多台服务器，很大的原因就是担心单个服务的蓝屏

18.1.1 什么是双机热备份

一般来说，多台计算机提供单个服务的环境，这些计算机必须处在一个完全相同（Identical）的环境下，为了操作方便，我们也会将这些计算机加入集群中（Cluster）。这里所谓的 Cluster 与 vCenter 中的 Cluster 原理相同，就是多台计算机的组合，只是在没有虚拟机的环境中，集群的单位就是一台台物理的服务器，因此操作的对象也是服务器。

1. 何谓双机热备

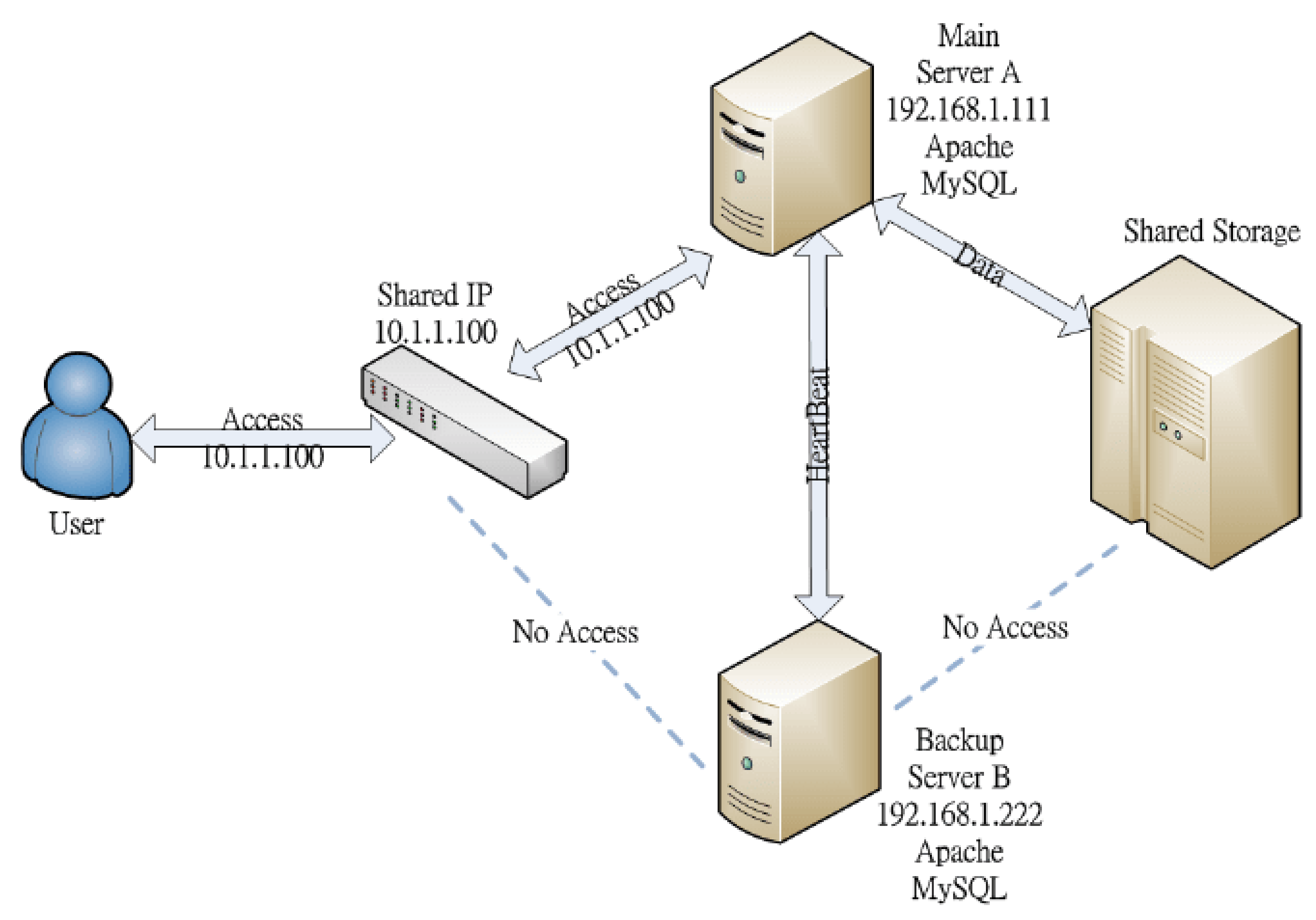
在不考虑提高性能的前提下，双机热备的原理相当简单，就是让单个服务随时都有两台以上的主机在线，但真正提供服务的只有单台主机，其他的机器都是单纯待命。这种称为 Active-Active 的集群环境，在虚拟机弹出之前是最完美的解决专案，对于大型的 Web 或是数据库的服务，的确能将停机时间降到最低。



▲ 双机热备一般称为 Failover Cluster

2. 双机热备的原理

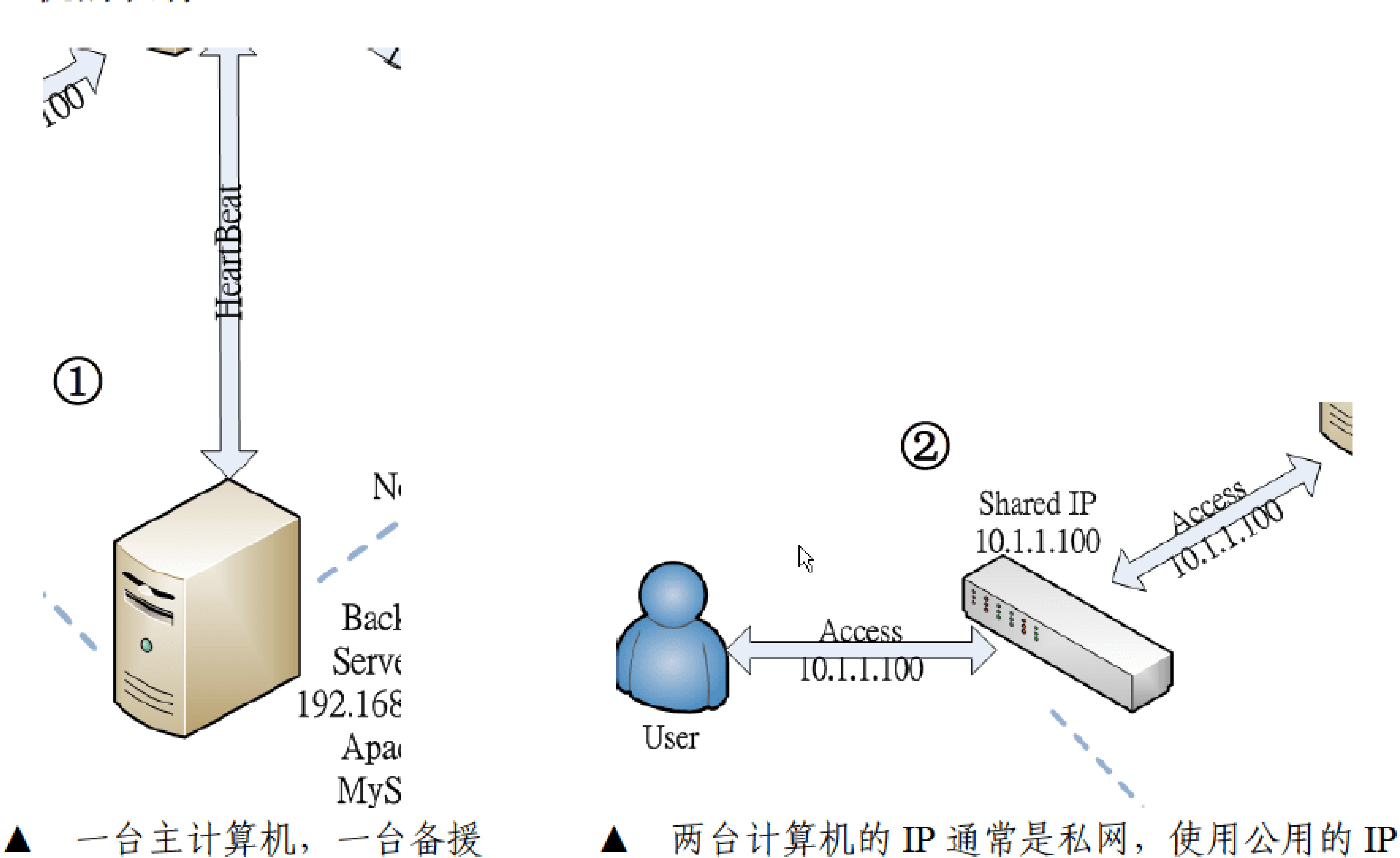
双机热备的原理也十分单纯，通常会有两台完全一样的机器，以及一个独立的存储设备。两台机器不但要完全一样，所有的操作系统、补丁、实用程序的版本都要完全一样。这两台机器所代表的通常是整个服务的逻辑层及表示层，而真正的数据层则是可写后端的存储设备，因此在缓存设计正确的情况下，数据是不会因为主机的蓝屏而丢失的。在两台以上的主机随时待命的情况下，发生主机蓝屏时，在第二台计算机上激活服务只是时间的问题，下面就是完整的示例。



▲ 这是标准双机热备的架构

▶ 双机热备的原理

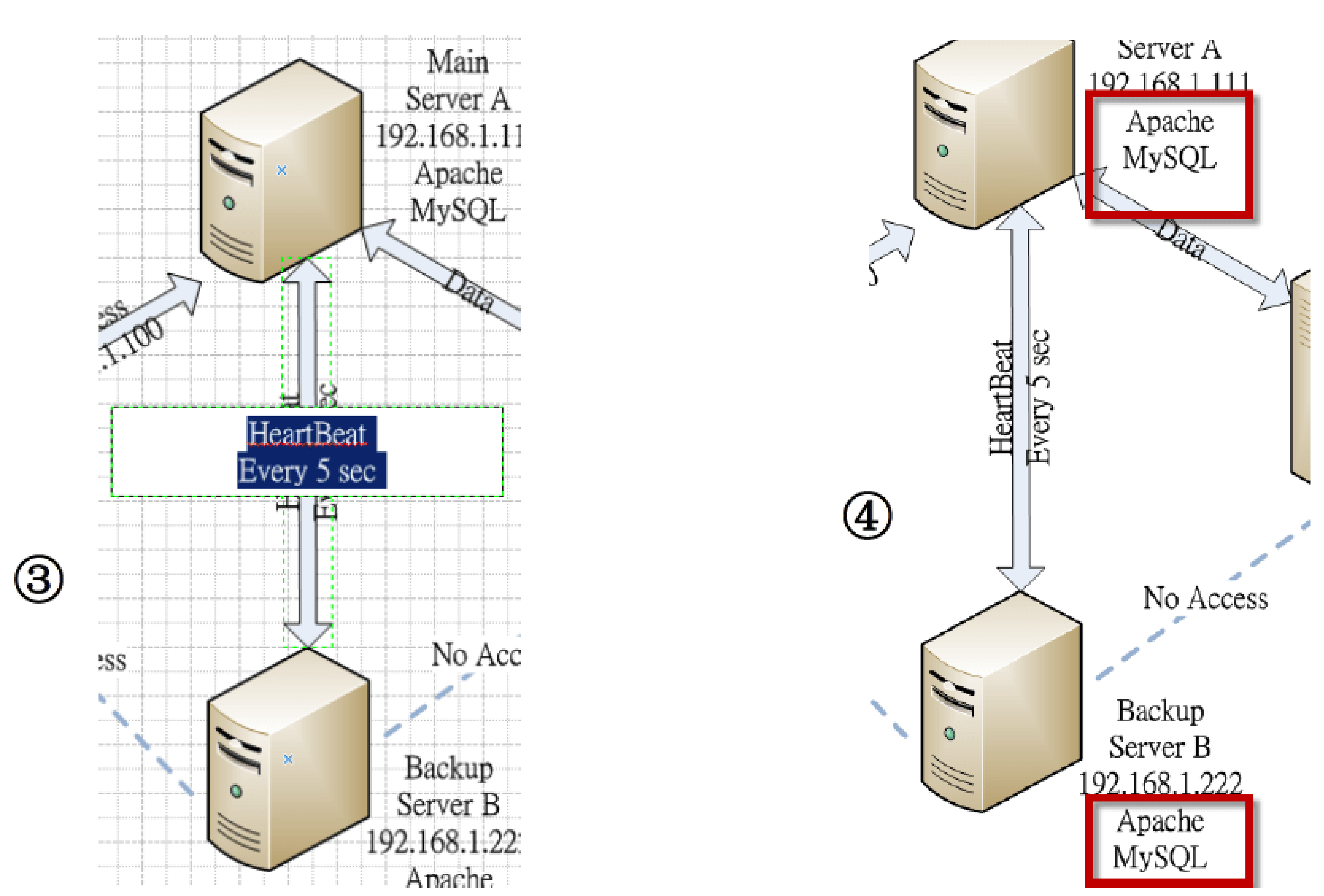
1. 两台计算机安装完全一样的软硬件，并且连结到一个后端的独立存储，我们姑且称为主计算机 A 和备援计算机 B。
2. 两台计算机对外的服务 IP 是一个代表性 IP，这个代表性 IP 事实上是连结到两个计算机的私有 IP 上。



▲ 一台主计算机，一台备援　　▲ 两台计算机的 IP 通常是私网，使用公用的 IP

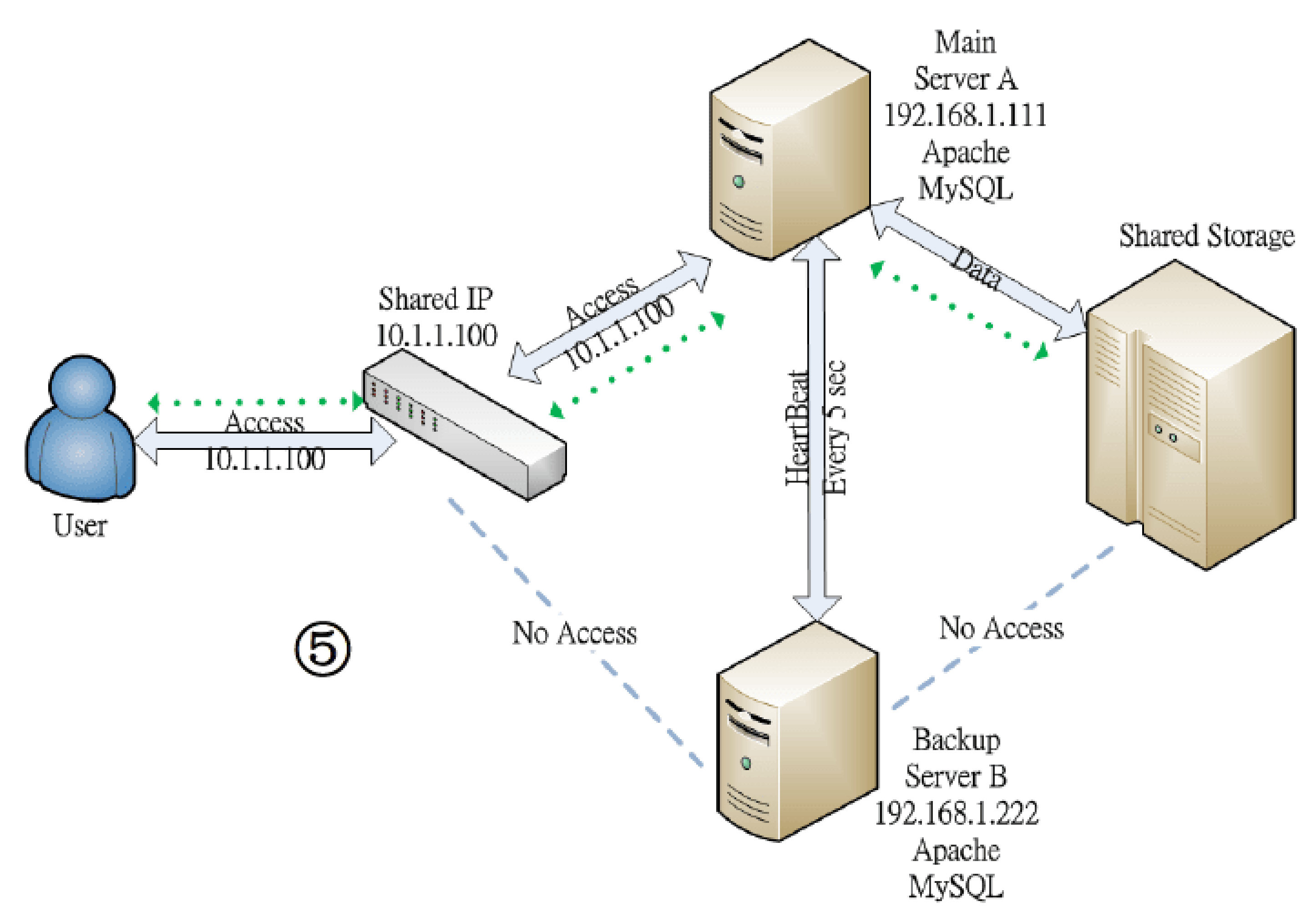
3. 两台计算机之间有一个心跳线（Heartbeat）来嗅探主机之间是否存活。遍历时间可以手动设置。

4. 两台计算机必须并发引导，并且运行完全一样的服务，连结到一样的网络存储，并且有完全一样的网络拓扑。



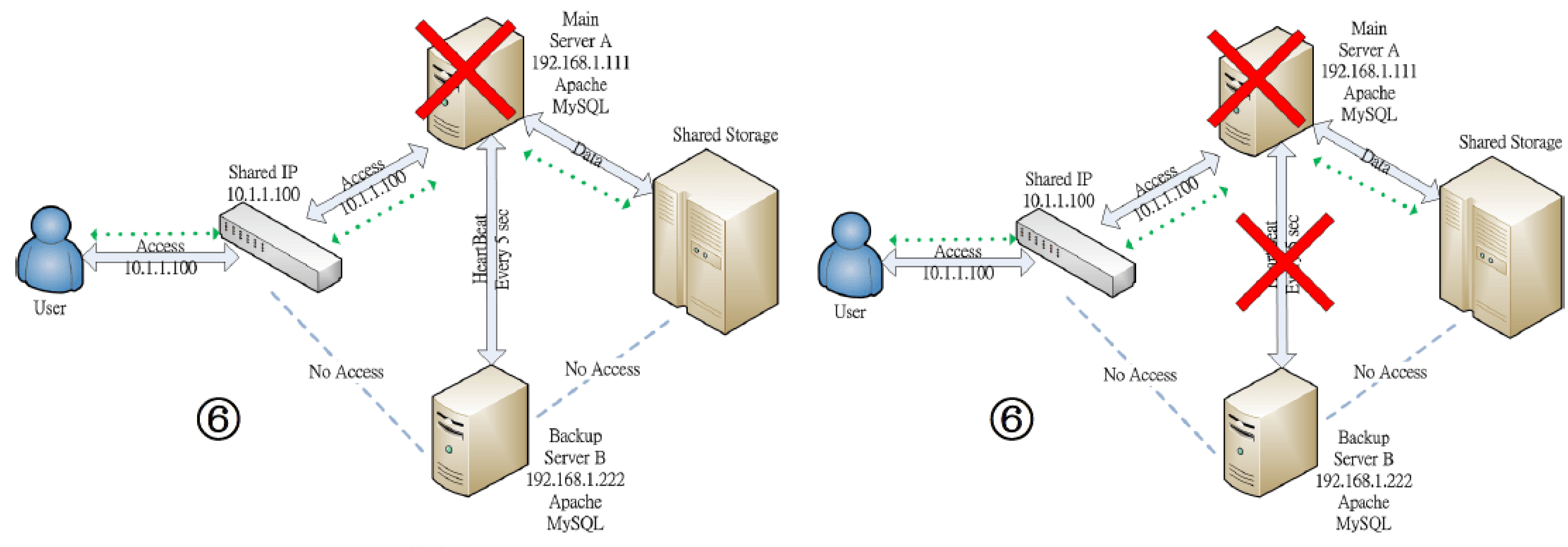
▲ 心跳线每隔一段时间遍历一次确认对方是否还活着 ▲ 两台计算机要并发引导，安装完全一样的服务

5. 当有服务生成时，仅有主计算机 A 进行读写，并且写到后端数据库，备援计算机 B 完全不作用。



▲ 常规情况的读写情况

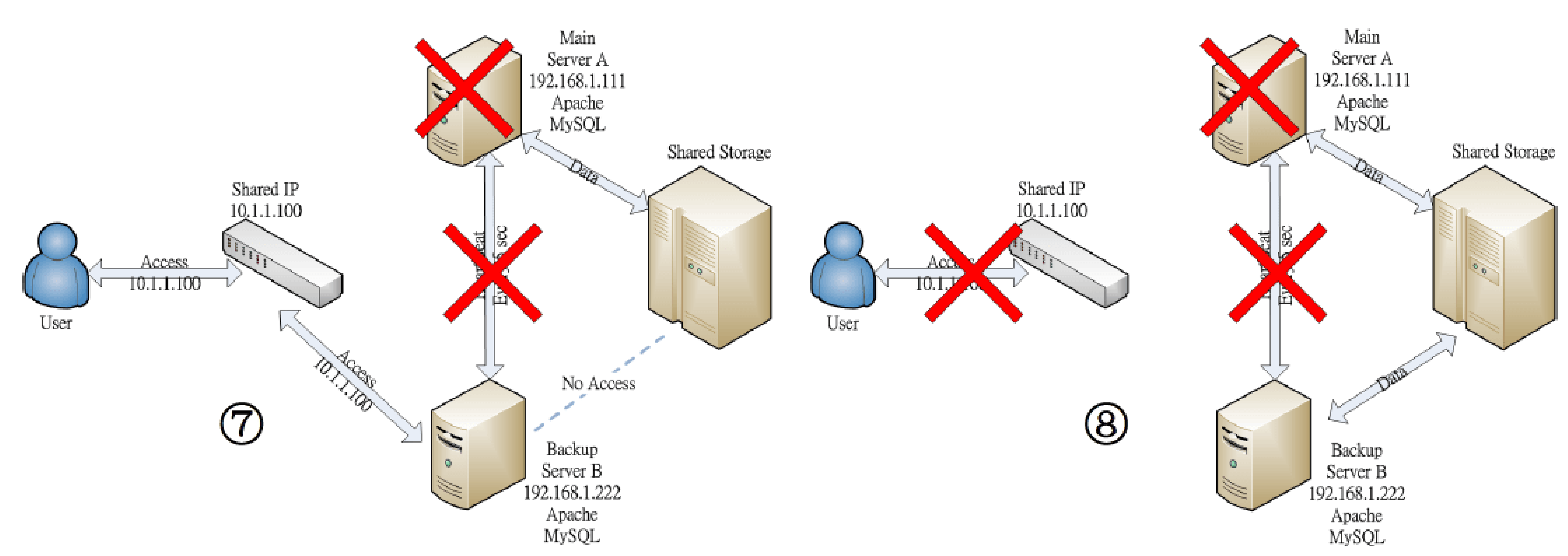
6. 当主计算机 A 蓝屏时，心跳线机会弹出中断，此时备援计算机 B 就会嗅探到服务中断，并且立即激活服务。



▲ 当主计算机 A 出问题时　　▲ 心跳线会发现 A 已经服务中止了

7. 此时代表 IP 也会发现服务中断，并且将服务从原来的主计算机 A 转向备援计算机 B。

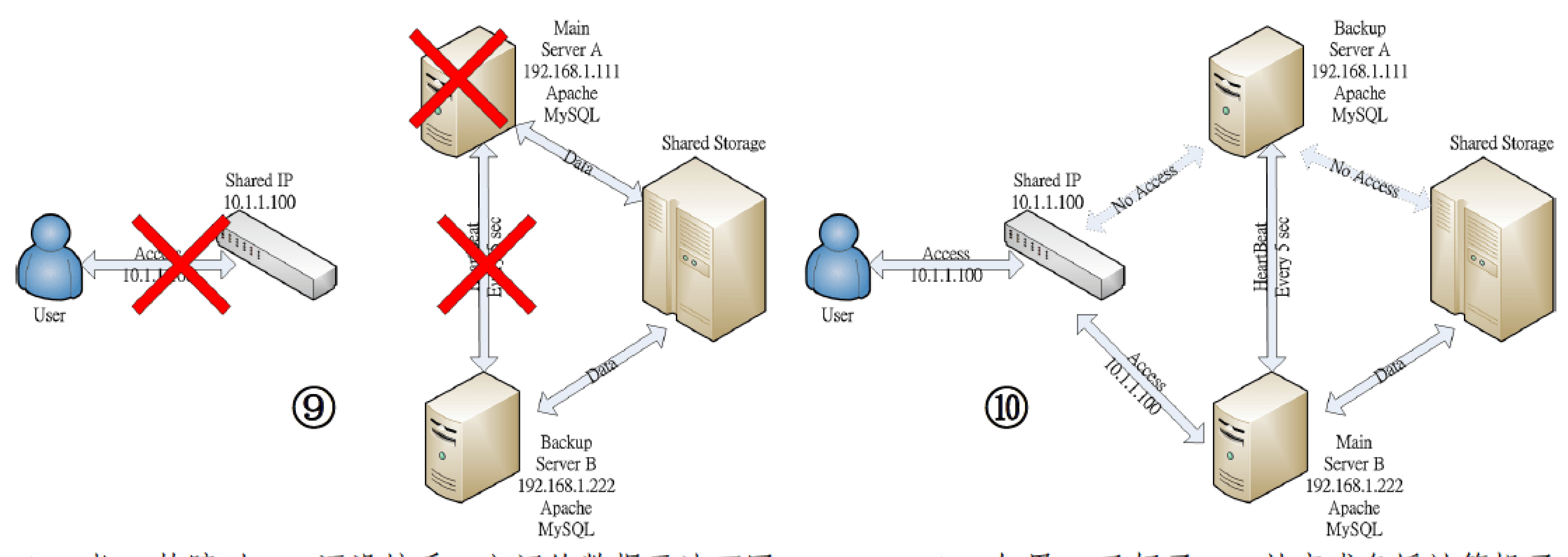
8. 此时所有的服务都会导向备援计算机 B，由于数据是放在独立的空间，因此数据并不会损失。



▲ 网络服务也发现 A 中断了，更换 IP 对应　　▲ 数据开始从 B 可写

9. 但在主计算机 A 故障之后，备援计算机 B 还没有激活前的数据并无法可写数据库，因此会丢失。

10. 此时备援计算机 B 会升级成主计算机，如果此时主计算机 A 修复落实重新加入集群，主计算机 A 将转换角色成为备援计算机，直至下一次备援计算机 B 弹出故障为止。



▲ 当 A 故障时，B 还没接手，之间的数据无法可写　　▲ 如果 A 又好了，A 就变成备援计算机了

18.1.2 双机热备的缺点

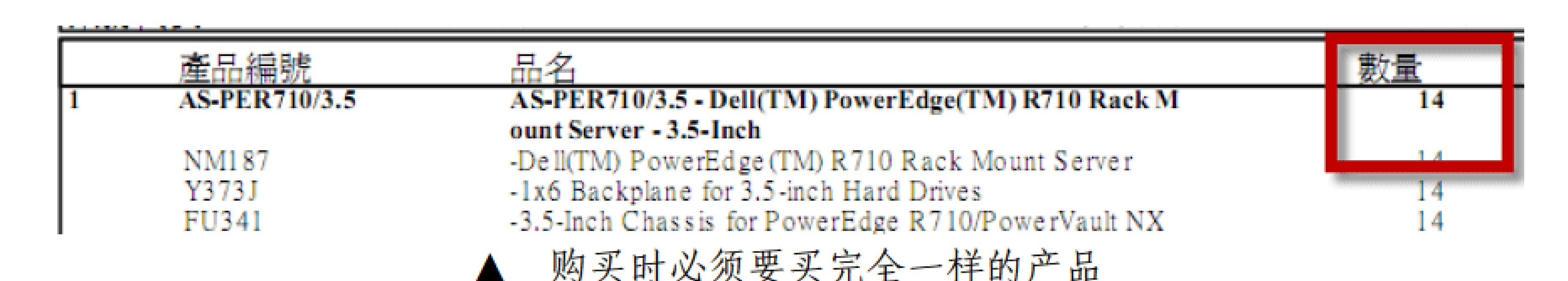
双机热备以当前来说，已经是非虚拟化解决专案之外最棒的做法了。但双机热备很明显有许多缺点，我们就来看看。

1. 高成本

以服务隔离的原则来说，每一台服务器上只跑一个服务已经算浪费了，双机热备是多台服务器跑一个服务，让双机热备的建置成本更高。一般来说，我们会使用双机热备的服务一定是很重要的，因此不能使用较阳春的服务器，那么多台高级服务器，会让整个架构的成本大量添加。

2. 制约性

由于双机热备需要完全一样的软硬件架构，因此当使用一段时间，真正有集群成员故障而需要补丁硬件时，很可能已经无法买到同样规格的产品了，如此一来集群的建置势必受到影响。



	產品編號	品名	數量
1	AS-PER710/3.5	AS-PER710/3.5 - Dell(TM) PowerEdge(TM) R710 Rack Mount Server - 3.5-Inch	14
	NM187	-Dell(TM) PowerEdge(TM) R710 Rack Mount Server	14
	Y373J	-1x6 Backplane for 3.5-inch Hard Drives	14
	FU341	-3.5-Inch Chassis for PowerEdge R710/PowerVault NX	14

▲ 购买时必须要买完全一样的产品

3. 复杂性

要进行双机热备，必须对存储、网络、冗余、集群等配置有很清楚的观念和实践经验，在设计不良的环境中，就算所有的软硬件都完全匹配，但也可能无法做出双机热备的平台。



▲ 有些数据库不支持多台计算机，设计上就会很麻烦

4. 蓝屏时间仍然无法避免

当主计算机 A 停机时，必须经过一段时间备援计算机 B 才能真正接手服务。在一个大量数据读写的环境，就算是几十秒的蓝屏时间，很可能已经有成千上万笔事务了，这对企业来说，无形的损失会远远大过有形的损失，更是企业质量的重大打击，因此就算使用双机热备，也是无法弥补这个缺点的。

18.2 真正零时差的双机热备：VMware Fault Tolerance

Fault Tolerance（简称 FT）充分利用虚拟机运作的原理。虚拟机运作在抽象化的硬件空间上，所有的资源都可随时生成、消灭、移转、克隆。VM 上的操作系统当然也可以立即生成、移转及克隆。如此一来，让两个完全一样的 VM 在内存之间进行录制及播放，真正做到 CPU 命令等级及内存比特等级的完全克隆/播放，这就是 VMware FT 的原理。

```
.encoding = "Big5"
config.version = "8"
virtualHW.version = "7"
scsi0.present = "TRUE"
scsi0.virtualDev = "lsilogic"
memsize = "512"
scsi0:0.present = "TRUE"
scsi0:0.fileName = "Ubuntu-cl1.vmdk"
ide1:0.present = "TRUE"
ide1:0.fileName = "auto detect"
ide1:0.deviceType = "cdrom-raw"
ethernet0.present = "TRUE"
ethernet0.wakeOnPcktRcv = "FALSE"
pciBridge0.present = "TRUE"
pciBridge4.present = "TRUE"
pciBridge4.virtualDev = "pcieRootPort"
pciBridge4.functions = "8"
pciBridge5.present = "TRUE"
pciBridge5.virtualDev = "pcieRootPort"
pciBridge5.functions = "8"
pciBridge6.present = "TRUE"
pciBridge6.virtualDev = "pcieRootPort"
pciBridge6.functions = "8"
pciBridge7.present = "TRUE"
pciBridge7.virtualDev = "pcieRootPort"
pciBridge7.functions = "8"
vmci0.present = "TRUE"
roamingVM.exitBehavior = "go"
displayName = "vaemon.com"
guestOS = "ubuntu"
nvram = "vaemon.com.nvram"
virtualHW.productCompatibility = "hosted"
ft.secondary0.enabled = "TRUE"
tools.upgrade.policy = "useGlobal"

extendedConfigFile = "vaemon.com.vmxf"

ethernet0.addressType = "generated"
uuid.location = "56 4d 54 4b cb 79 28 1e-4c 40 da 6e b9 97 46 82"
uuid.bios = "56 4d 7e 74 3d 83 4c 22-c8 94 7c 3b e2 d5 34 75"
scsi0:0.redo = ""
vmotion.checkpointFBSize = "20512768"
pciBridge0.pciSlotNumber = "17"
```

▲ 由于 VM 都是以文件的方式存在，因此要生成一个 VM 就是克隆文件

18.2.1 VMware FT 是内存和 CPU 的 RAID 1

VMware FT 使用了一个称为 vLockstep 的技术，这个技术在 2006 年的 VMware Workstation 6.0 上就弹出了。vLockstep 主要就是撷取一个 VM 上的命令，并且在另一个 VM 上完全运行。下面我们就来看看 vLocksetup 的原理。

1. 重要命令的录制/回放（Deterministic Record/Replay）

在 vLockstep 上，将命令分为重要（Deterministic）及非重要（non-Deterministic）两种。重要命令的录制/回放一直很难达成，主要的原因就是外部的键入如网络报文、鼠标键盘操作、磁盘 IO 等落实事件，通常会在非接轨的模式下操作，如此一来会改变原来命令的运行路径。重要命令录制播放功能，可以将非重要命令录下来，并且在播放时，在同一个运行点注入。这个方式可以省下大量的内存和 CPU 运算资源，也是 VMware FT 能在 vSphere 中推出的重要原因。

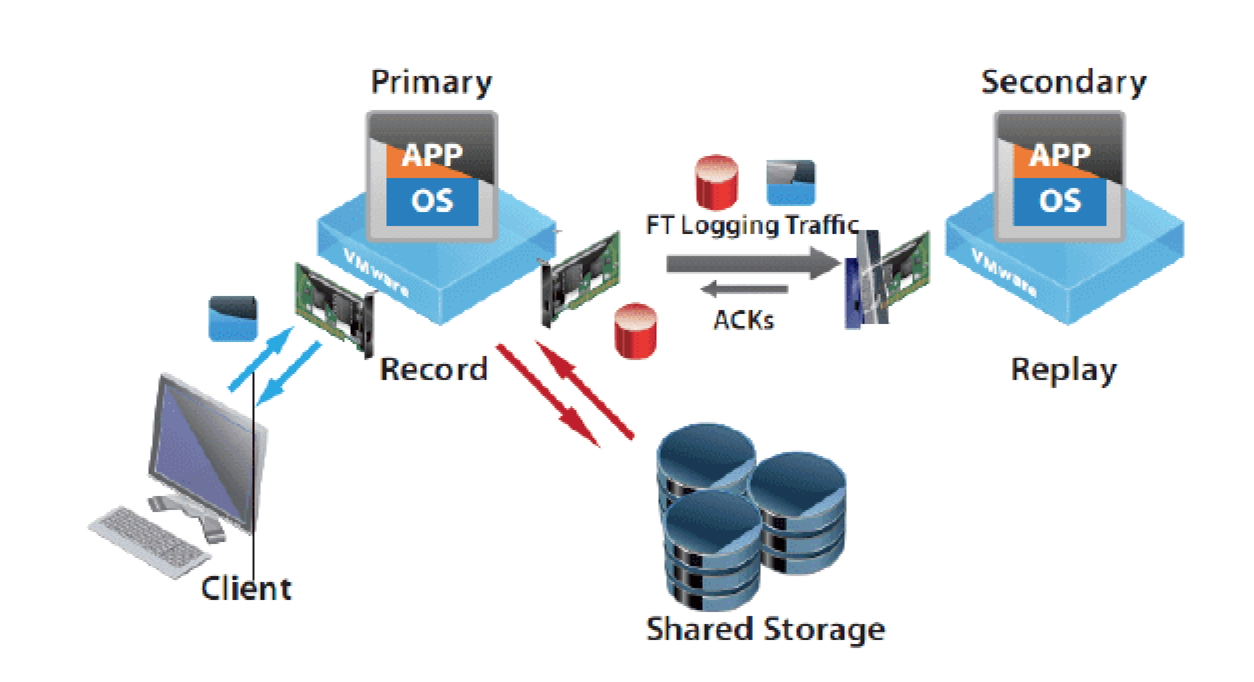


▲ VMware FT 使用了 vLockstep，因此可以在副本 VM 插入命令

> **注意**
>
> **需要 CPU 的支持**
>
> 由于 VMware FT 是 CPU 等级的运算功能，因此需要 CPU 的支持才能进行。VMware 和两大 CPU 厂家 AMD 和 Intel 合作多年，已经在大部分的产品中支持 FT 命令了。但当前只能用在单 CPU 的 VM 中，如果你在 VM 中给定了多 CPU 就无法使用 FT 功能了。这是因为多 CPU 除了录制播放之外，还得考虑到内存的共享及顺序问题，会让整个运作的复杂性更高。我们会在本章稍后给出支持的 CPU 产品列表。

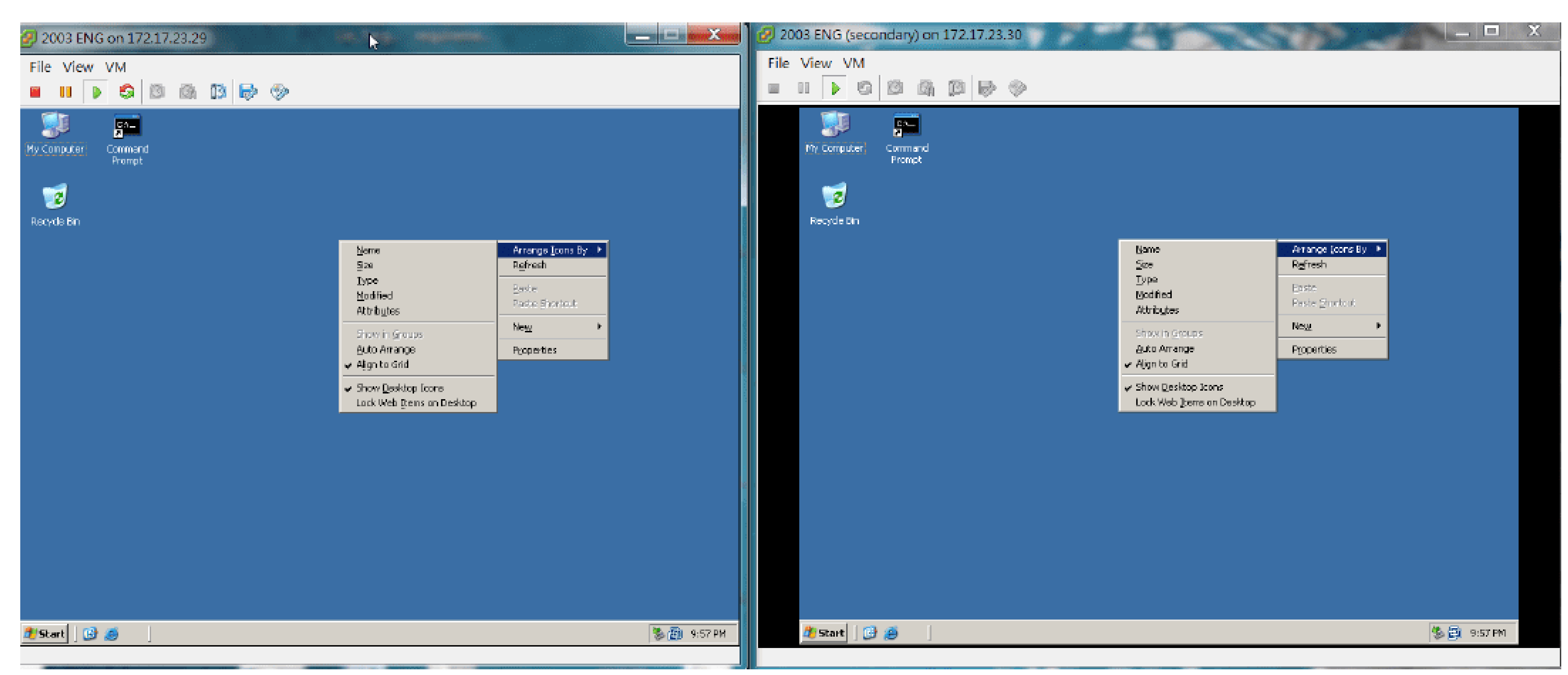
2. VMware FT 是如何办到的

VMware FT 是 vLockstep 中的重要命令录制/播放功能，其原理十分简单，就是让两台物理机上的 ESX 各有一个 VM 的正本和副本。这两个 VM 的正副本是完全一样的，不管正本的 VM 做什么动作，都会立即在 CPU 的命令集层级和内存的比特层级立即克隆一份到副本上。任何对正本 VM 的动作也会并发提交到副本 VM 上。



▲ 在两台物理 ESX 主机上各有一个 VM 的正本及副本

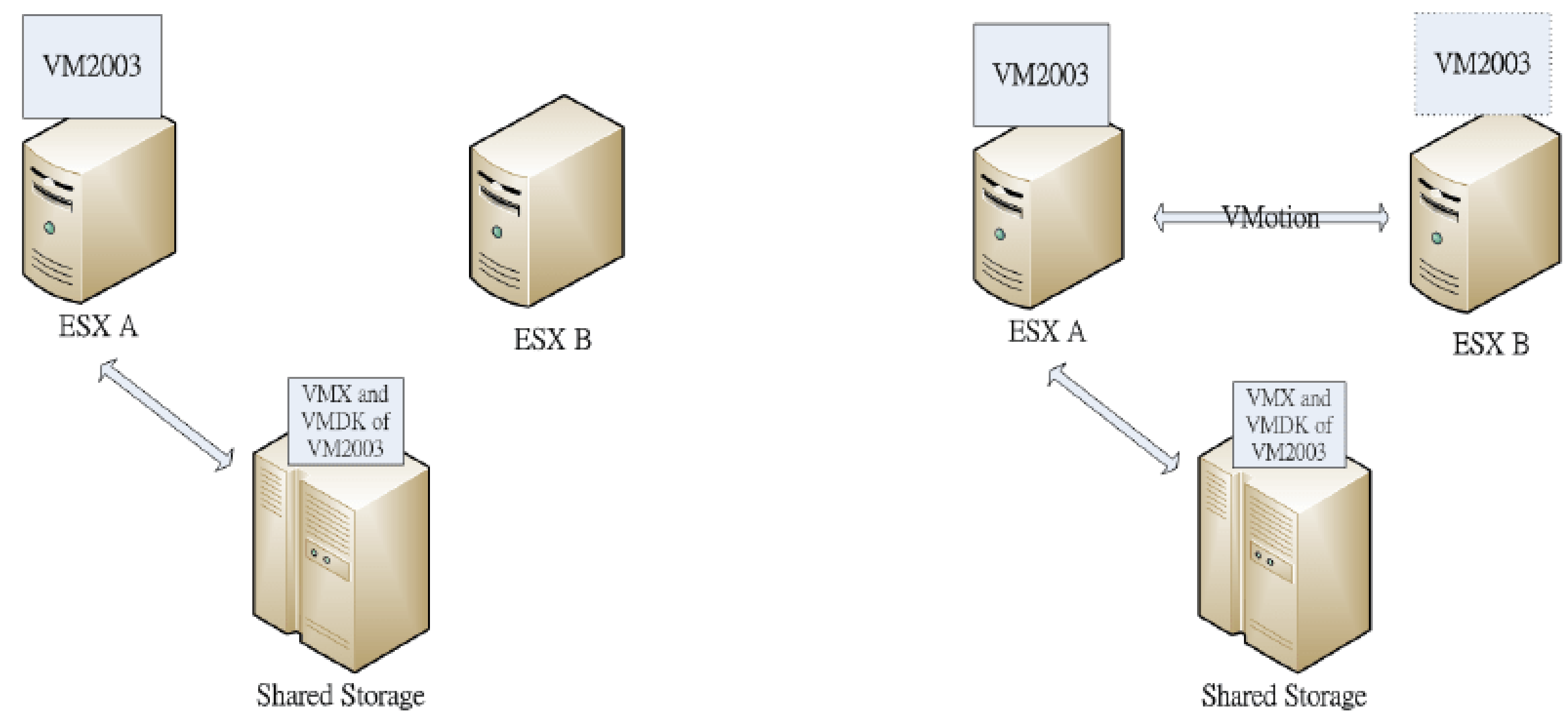
具体来说，当正本的 VM 激活后，VMware FT 会在集群中的另一台 ESX 主机上，利用 VMotion 来实时克隆正本 VM 的内存状态。当 VMware FT 激活之后，副本的 VM 将会利用 vLockstep 的功能，亦步亦趋地完整克隆正本的 CPU 命令。下面我们就来看看运行原理。



▲ 激活之后的 VM 副本会完全和正本接轨

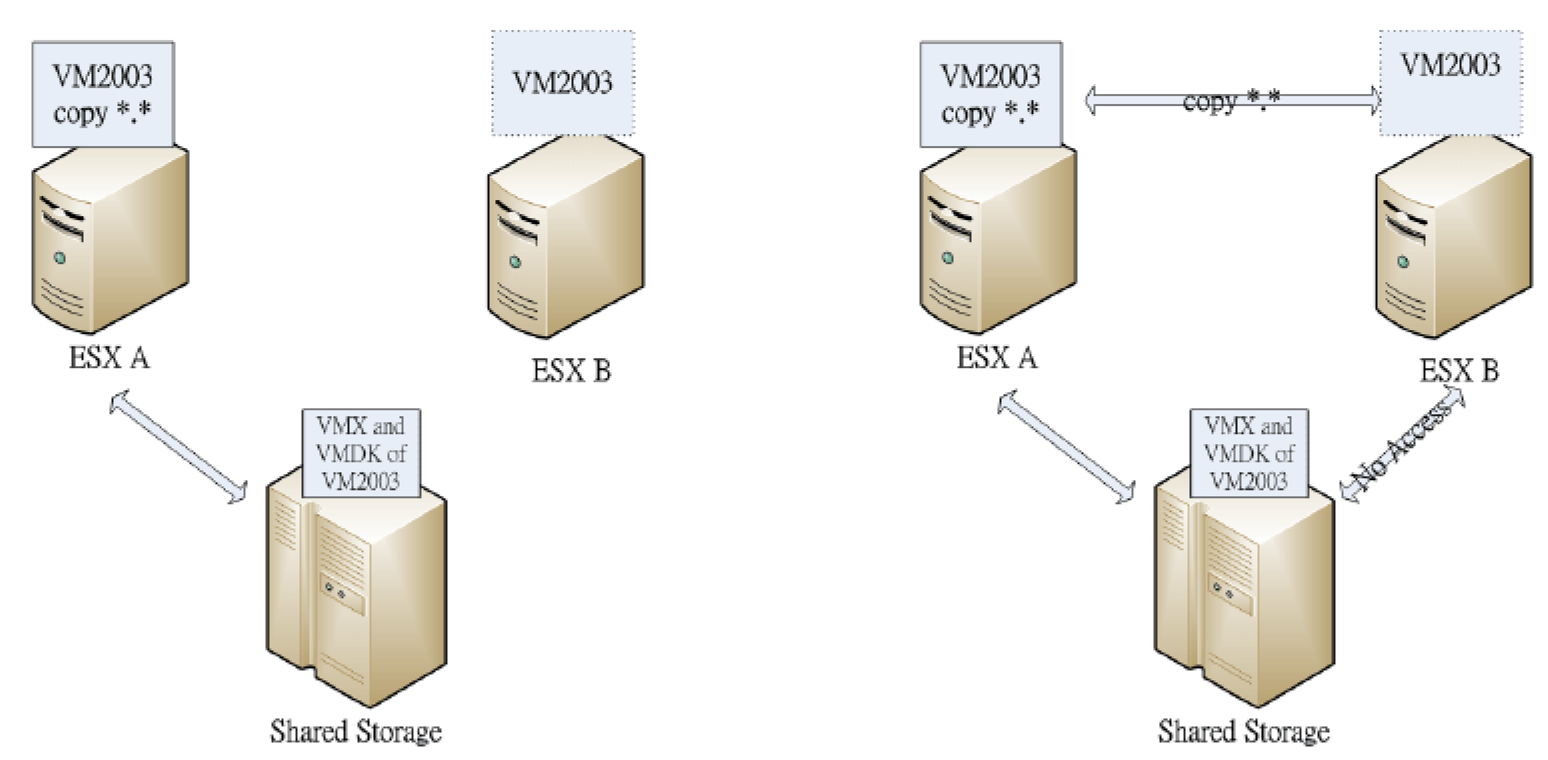
▶ VMware FT 的运行原理

1. 正本和副本的 VM 是由各自的 ESX 主机所总控。举例来说，正本运行在 ESX 物理机 A，而副本在物理机 B。



▲ 本来只有一个正本 VM　　▲ 激活 FT 后会使用 VMotion 在第二台物理机上创建一个副本

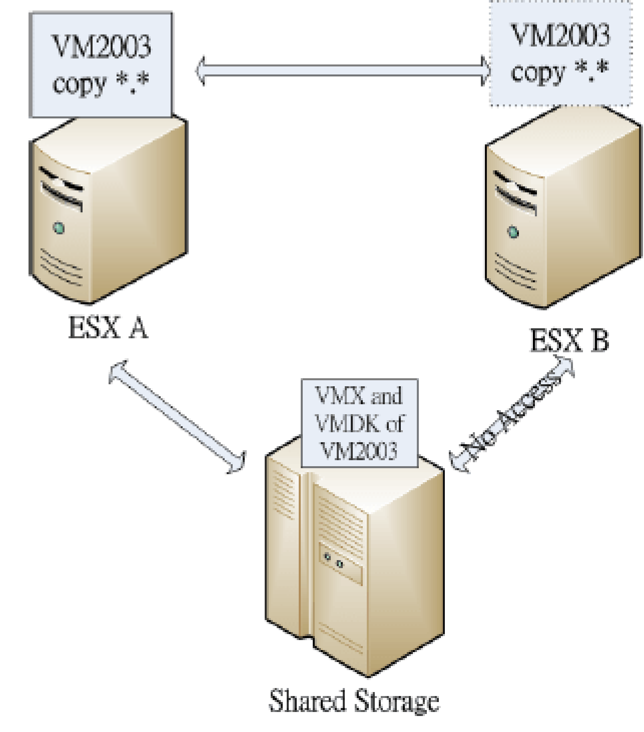
2. 正本 VM 所接受或运行的任何 CPU 命令，会被物理机 A 的 ESX 录制下来，并且立即提交到物理机 B 的 ESX 上。



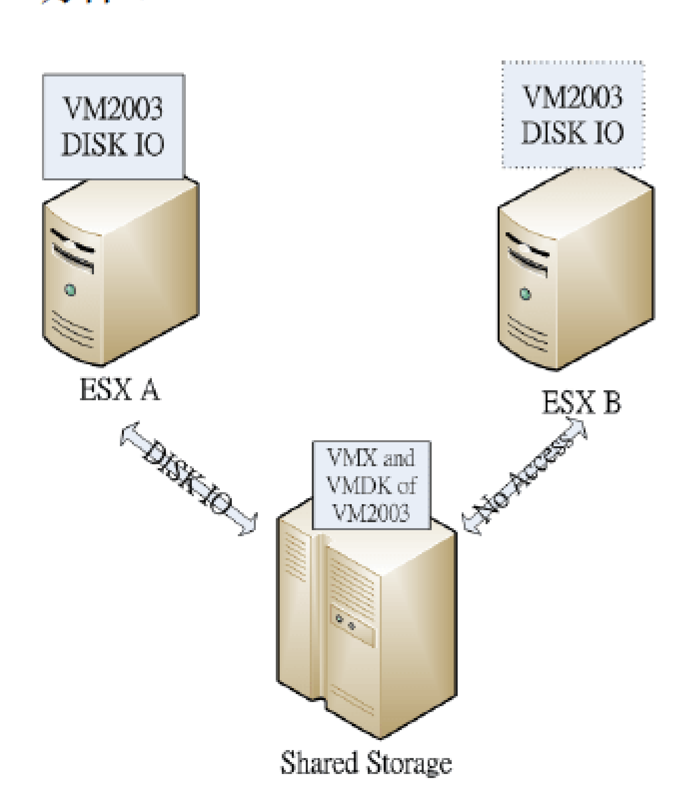
▲ 在正本 VM 进行的任何动作都会被克隆下来▲ 立即提交到副本的 VM 插入命令，让副本的 VM 也动作

3. 物理机 B 将这些命令在不同的运行点，直接注射到副本 VM 中。

4. 正本和副本的 VM 使用同一个虚拟磁盘（在独立存储上）。但所有的 I/O 操作只会在正本 VM 上，副本的 VM 不会有任何 IO 动作。

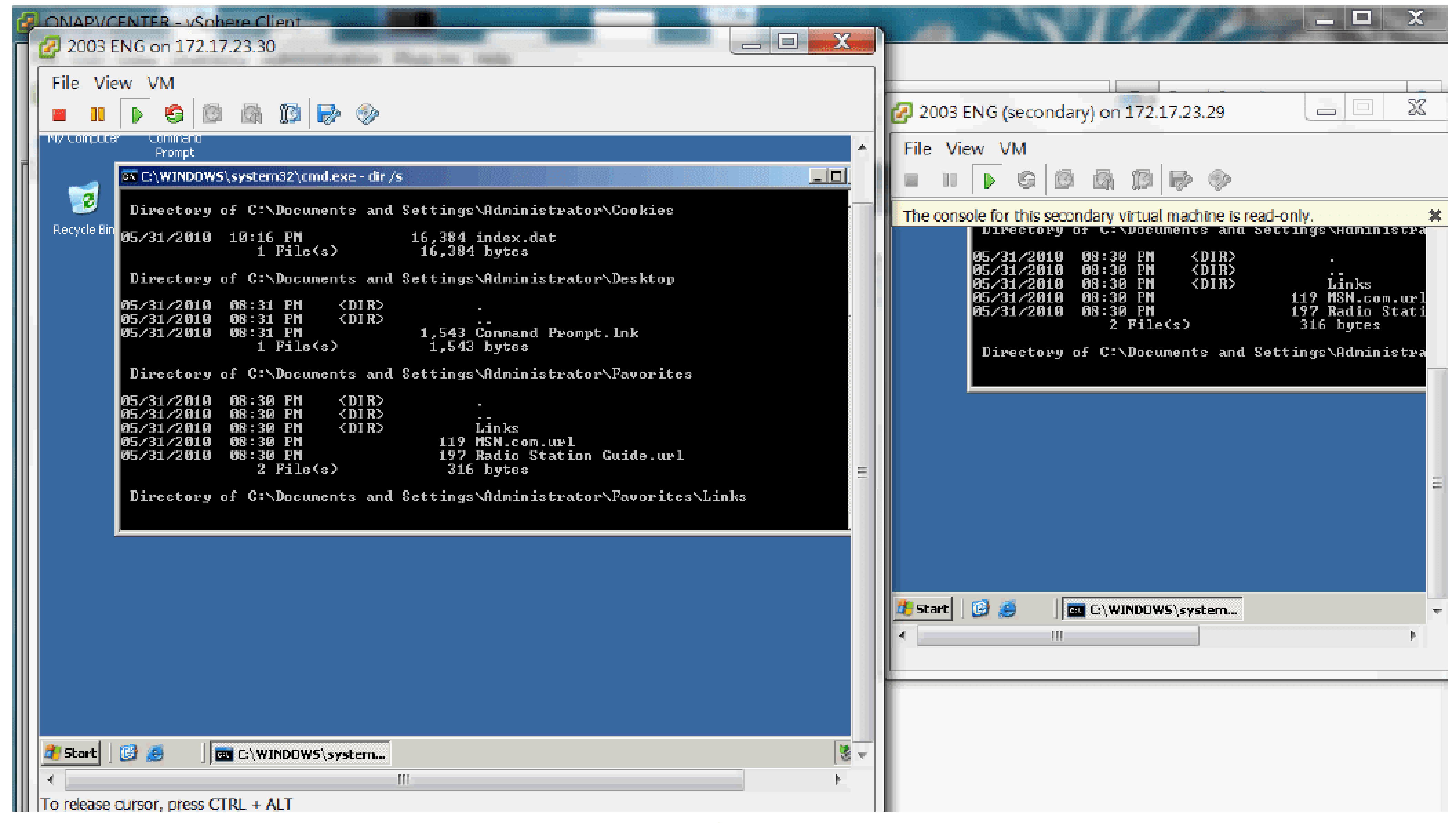


▲ 副本 VM 立即运行命令



▲ 有任何 IO 动作时，会可写同一个文件

5. 在具体观察时，你会发现两个 VM 动作完全一样，好像照镜子一般，但热点是，副本的 VM 是处于只读状态，你无法在副本的 VM 上进行任何动作。

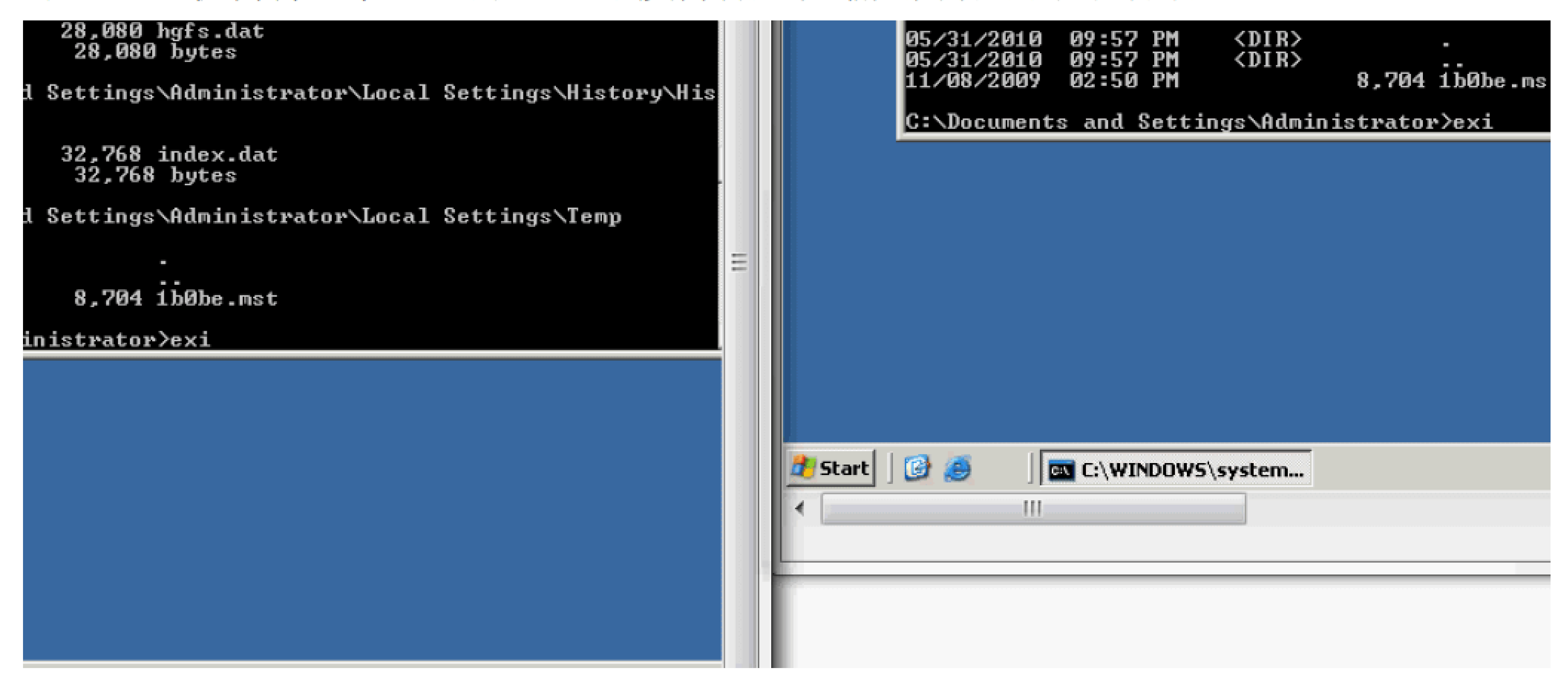


▲ 两者动作完全一样

3. 正本和副本 VM 的时间差

VMware FT 尽可能做到两个 VM 的动作完全一样，但由于是录制/播放的做法，因此一定会有一个时间差，这个时间差称为 vLockstep interval。我们在本章稍后的实践中可以看到这个数值。

vLockstep interval 的时间长短，决定于正副本 VM 所在 ESX 主机的 CPU 及内存资源。如果资源足够，两者之间的差距会在几分之一秒，但如果当副本 VM 的动作赶不上正本 VM 时，正本所在的 ESX 主机会将正本 VM 的 CPU 速度降低，以缩短两者之间的时间差。



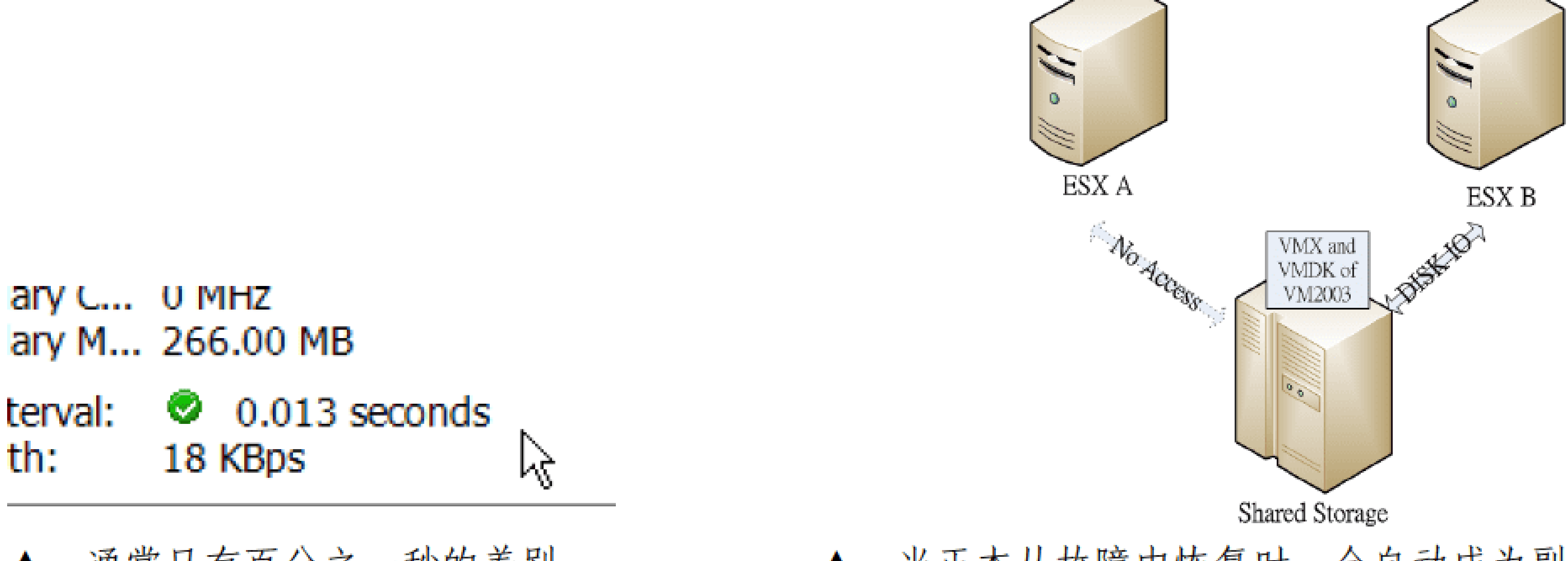
▲ 正副本 VM 之间会有一个时间差，但通常都很短

4. VMware FT 动作时机

VMware FT 的激活时机就是正本 VM 所在的 ESX 主机故障时。当正本 VM 所在的 ESX 故障时，副本的 VM 会立即从“只读”状态变成常规状态，并且立即升级成“正本”VM，开始接手

所有的动作。由于正本和副本 VM 之间的时间差通常在几分之一秒甚至是几百分之一秒，因此之间的服务中断时间几乎可忽略不计，用户根本不会察觉正本 VM 所在的 ESX 主机已经停止。

当正本 VM 所在的 ESX 从故障中恢复时，此时正本 VM 已经转到了 ESX B 上，因此原来的 ESX A 会被降级成副本 VM，而副本 VM 也会激活。在一个较完整的 VMware FT 环境中，可以准备两台以上的副本 VM，以达到更安全的架构。



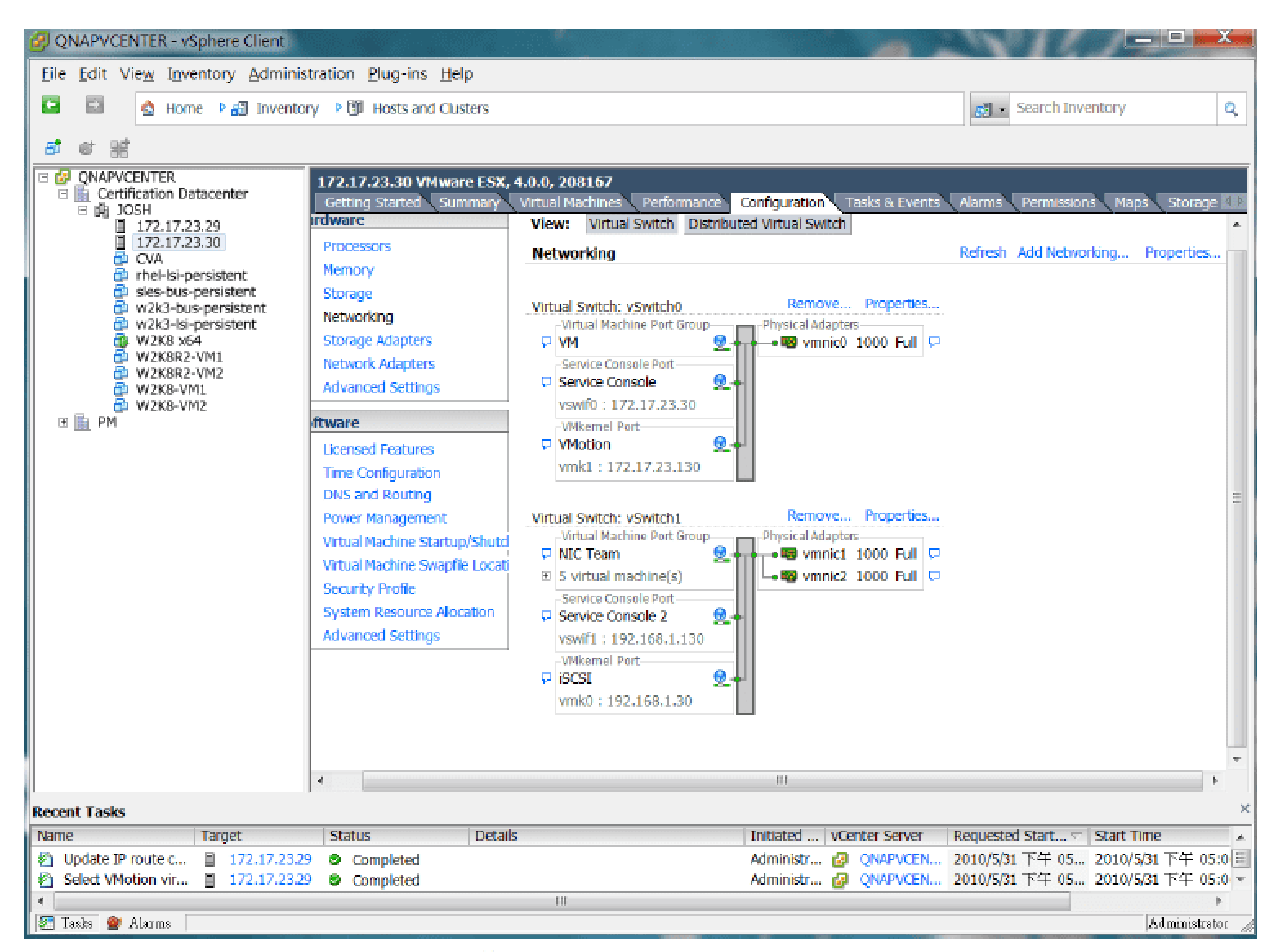
▲ 通常只有百分之一秒的差别　　▲ 当正本从故障中恢复时，会自动成为副本

18.2.2 VMware FT 解决的问题

VMware FT 解决了所有双机热备无法解决的问题，更利用 vSphere 的架构解决了集群架构的复杂性，我们就来看看。

1. 降低成本

vSphere 架构本来就是多台服务器，因此在原有的服务器上架设 VMware FT 功能，并不需要新建任何机器设备。但要注意的是，如果你想使用 VMware FT 功能，必须在较高级版本的 vSphere 上才有提供，也必须支付授权费用。



▲ 使用原有的 vSphere 集群

2．弹性较大

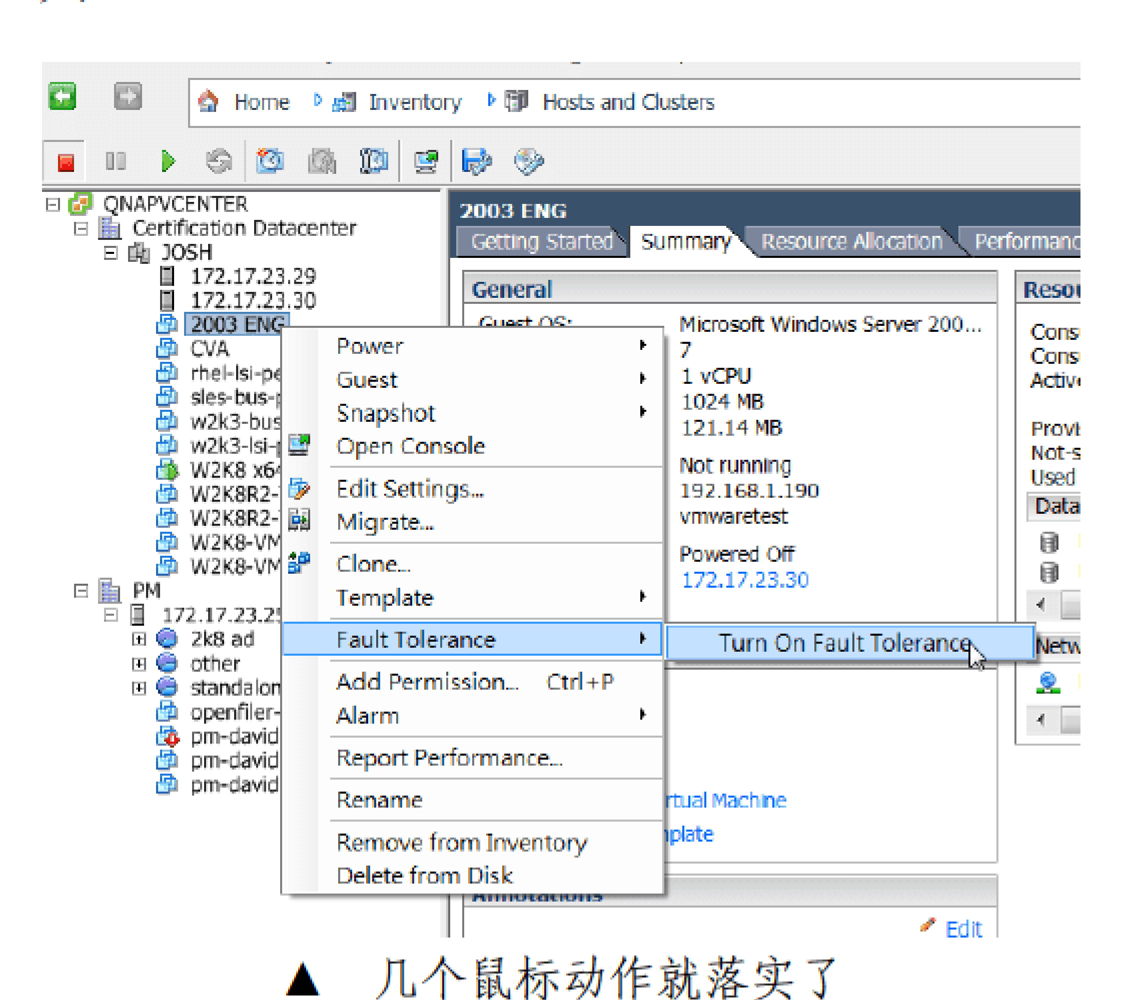
只要是匹配 VMware FT 的 CPU 列表上的产品都可以运行 VMware FT 功能，因此你不需要购买特定品牌的服务器或存储设备，只要使用兼容的 CPU 即可。如此一来，可以在集群中使用不同品牌的服务器，对预算总控较严格的企业来说是相当方便的。



▲ 本书生产环境的 FT 使用了两种不同品牌的服务器，HP 和 DELL

3．环境较单纯

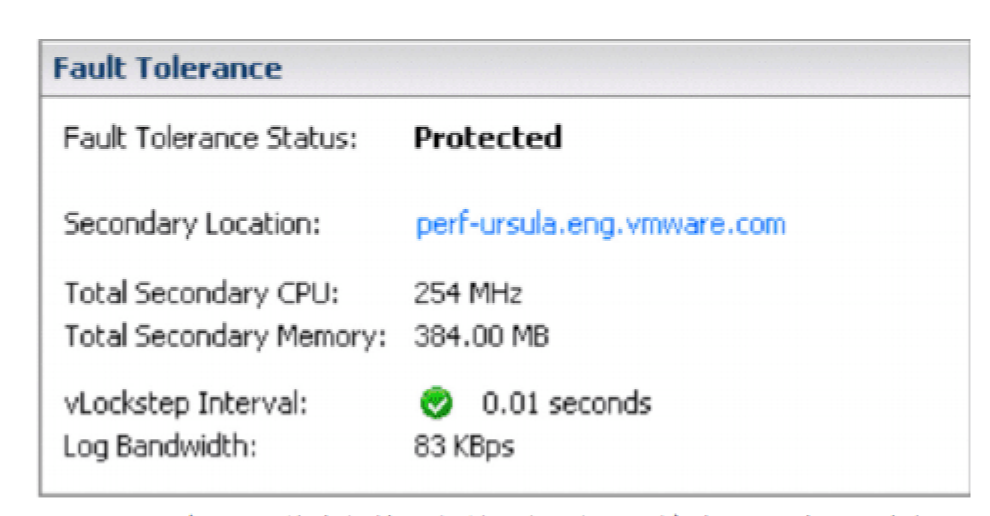
VMware FT 的配置只需几个鼠标的动作即可落实，由于 VMware FT 架构在 VMotion 上，因此只要你的 vCenter 环境已经将 HA 和 DRS 功能激活了，激活 FT 功能就更简单了。这可以大量降低配置及布建环境的工夫。



▲ 几个鼠标动作就落实了

4．几乎没有停滞时间

VMware FT 的停滞时间就是 vLockstep interval，因此这短短的几十分之一秒对系统造成的影响的确小之又小。试想在一个复杂的 Web/SQL 环境，如果两台双机热备的主机能将服务停止时间降到贴近 0，那么用户几乎不会感觉到停顿，一个永远不会蓝屏的环境就俨然成型了。



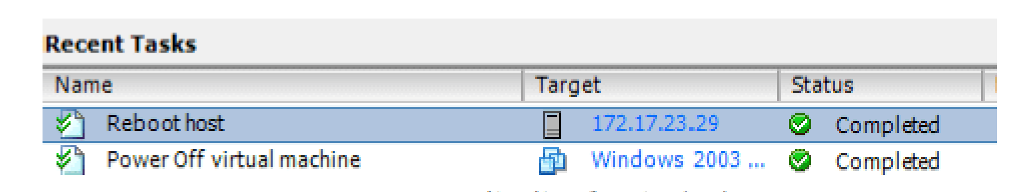
▲ 每一种操作系统的时间差都不太一样

18.2.3　VMware FT 和 HA 的异同

很多读者一定会问，既然 VMware 已经有 HA 了，为何还要有 FT 呢？事实上，HA 和 FT 的防护等级根本不在同一层次。我们就来看看两者的异同及使用时机。

1．HA 的中断时间较长

VMware HA 最大的特色就是使用 VMotion，当某一台 ESX 主机蓝屏时，能将其上的 VM 利用 VMotion 移动到没有蓝屏的 ESX 主机上。但使用 HA 的 VM 最大的问题就是必须重启。

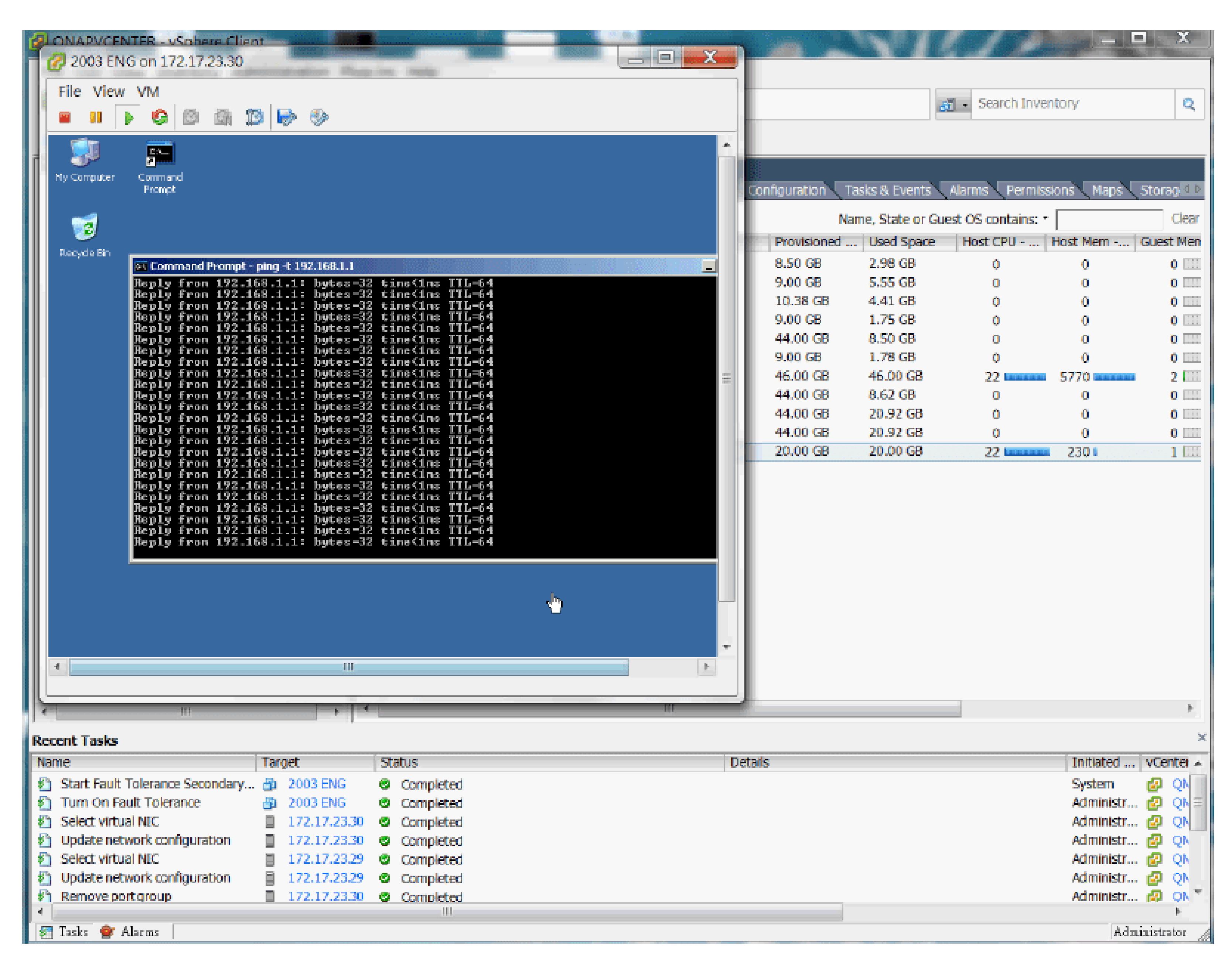


▲　HA 常常需要重启

举例来说，当我们使用某个 Web 服务时，如果这个 Web 服务所在的 VM 其物理 ESX 主机蓝屏了，此时这个 Web 服务会立即停止，直到 HA 服务将这台 VM 移到另一台物理 ESX 主机上引导落实后，Web 服务才能恢复。这个时间通常会花上数分钟，虽然服务能够延续，但中间已经有数分钟的 Downtime 了。

2．FT 的中断时间几乎可不计

在同样的情况下，如果我们使用 FT，那么中断的时间只有几十分之一秒，所占用的资源也只是原来就有的主机，因此在系统资源较充裕的环境，笔者强烈建议使用 FT 来取代 HA，可以拥有更安全的架构。



▲　使用 ping 会发现根本没有中断

18.3　实施 VMware FT 的前提

虽然在 vSphere 中出现了 FT 这么好的功能，但并不是每一台机器都能完整使用 FT 功能，而

FT 功能也出现在特殊付费版本的 vSphere 中。我们就来看看要实施 FT 功能的基本条件。

▶ 支持 FT 功能的 vSphere 版本

- vSphere 进阶版
- vSphere 企业版

18.3.1 支持 FT 的 CPU 列表

要支持 FT 功能，必须 CPU 支持 FT 命令才行。大部分较新的 AMD 或 Intel CPU 都已经支持 FT 功能了。但不同版本的 CPU 支持的 FT 功能不尽相同，有些需要在 VM 关机时激活，我们就来看看支持的 CPU 列表。

1. AMD CPU 系列

AMD 支持 FT 功能的 CPU 为第三代的 Opteron 家族，包括下列产品：

▶ AMD 支持 FT 的 CPU 列表

- 13xx and 14xx Series
- 23xx and 24xx Series （DP）
- 41xx Series*
- 61xx Series*
- 83xx and 84xx Series （MP）

2. Intel CPU 系列

支持 FT 的 Intel 产品较多，当前支持 FT 的 Intel CPU 包括了 45nm 的 Core2 内核以及 i7 内核的产品，下面就是列表。

▶ 支持 FT 的 Intel CPU 列表

Intel Xeon based on 45nm Core 2 类别：

- 31xx Series
- 33xx Series
- 52xx Series （DP）
- 54xx Series
- 74xx Series

Intel Xeon based on Core i7 类别：

- 34xx Series （Lynnfield）*
- 35xx Series
- 55xx Series

3. 支持的客户端操作系统

视不同的 CPU 产品而定，支持 FT 的客户端操作系统也不尽相同，一共有三种不同的情况。

▶ 客户端操作系统和 CPU 的组合

- 支持：即 VM 中的操作系统在引导时即可激活 FT 功能。
- 支持/关机：VM 中的操作系统必须在关机时才能激活 FT 功能。
- 不支持：VM 中的操作系统和 CPU 的组合不支持 FT 功能。

我们就来看看支持列表。

表 1　客户端操作系统的 FT 支持列表

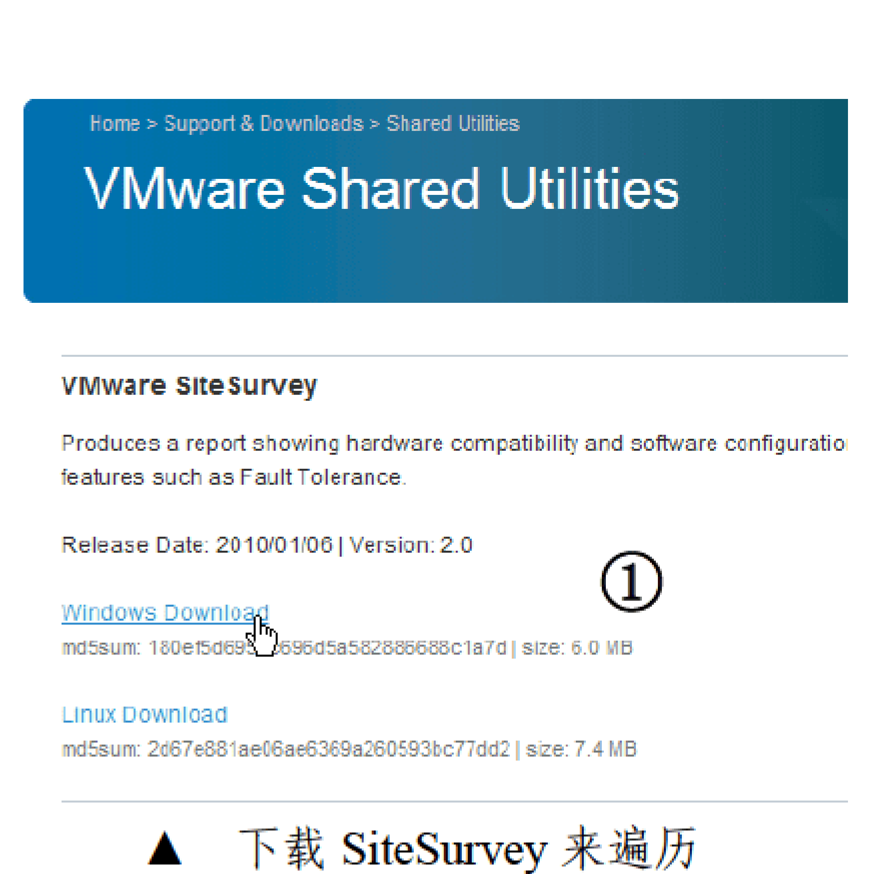
客户端操作系统	Intel 45nm Core2 架构	Intel Core i7 架构	AMD 3rd G Opteron
Windows 7	支持	支持/关机	支持/关机
Windows Server 2008	支持	支持/关机	支持/关机
Windows Vista	支持	支持/关机	支持/关机
Windows Server 2003 X64	支持	支持/关机	支持/关机
Windows Server 2003 X86 SP2	支持	支持/关机	支持/关机
Windows XP X64	支持	支持/关机	支持/关机
Windows XP X86	支持	支持/关机	不支持
Windows 2000	支持/关机	支持/关机	不支持
Windows NT 4.0	支持/关机	支持/关机	不支持
Linux（所有 ESX 支持的版本）	支持	支持/关机	支持/关机
Netware Server	支持/关机	支持/关机	支持/关机
Solaris 10 （64 比特）	支持	支持/关机	支持/关机
Solaris 10 （32 比特）	支持	支持/关机	不支持
FreeBSD（所有 ESX 支持的版本）	支持	支持/关机	支持/关机

4．使用 SiteSurvey 工具来判断

VMware 提供了一个工具让你判断当前在 vCenter 主机下管理的 ESX 主机是否能支持 FT 功能，称为 SiteSurvey。你只要下载安装，即可弹出 HTML 的报表来检视当前使用的机器是否能支持 FT。我们就来看看 SiteSurvey 的使用手段。

▶ 使用 SiteSurvey 来检视主机是否支持 FT

1. 先去 VMware 的网站下载 SiteSurvey 软件，你可选择 Windows 或 Linux 版本。
2. 下载回来后直接安装，SiteSurvey 就像任何一个 Windows 实用程序一样，安装时只要一直单击 Next 按钮即可落实安装。

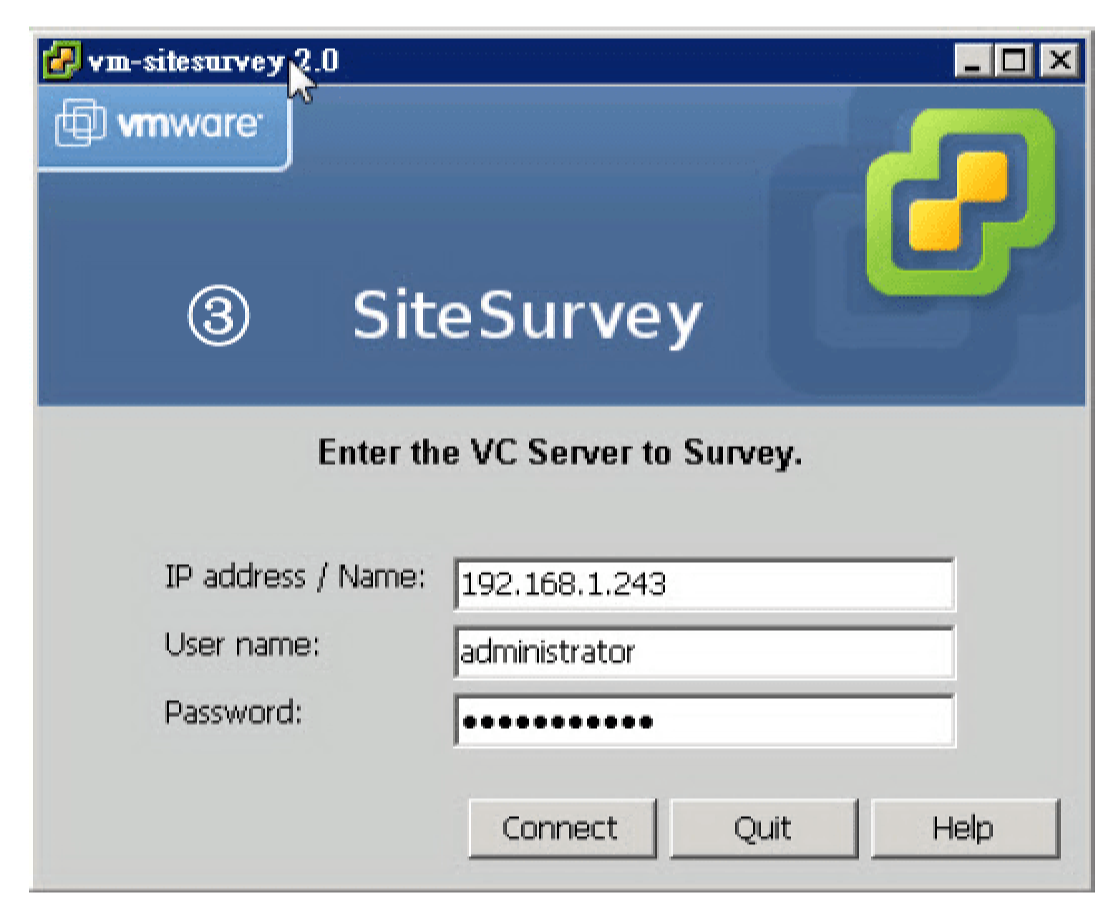


▲　下载 SiteSurvey 来遍历

▲　直接安装即可

3. 安装落实后，直接运行 SiteSurvey。键入 vCenter 的主机 IP 及账号口令。

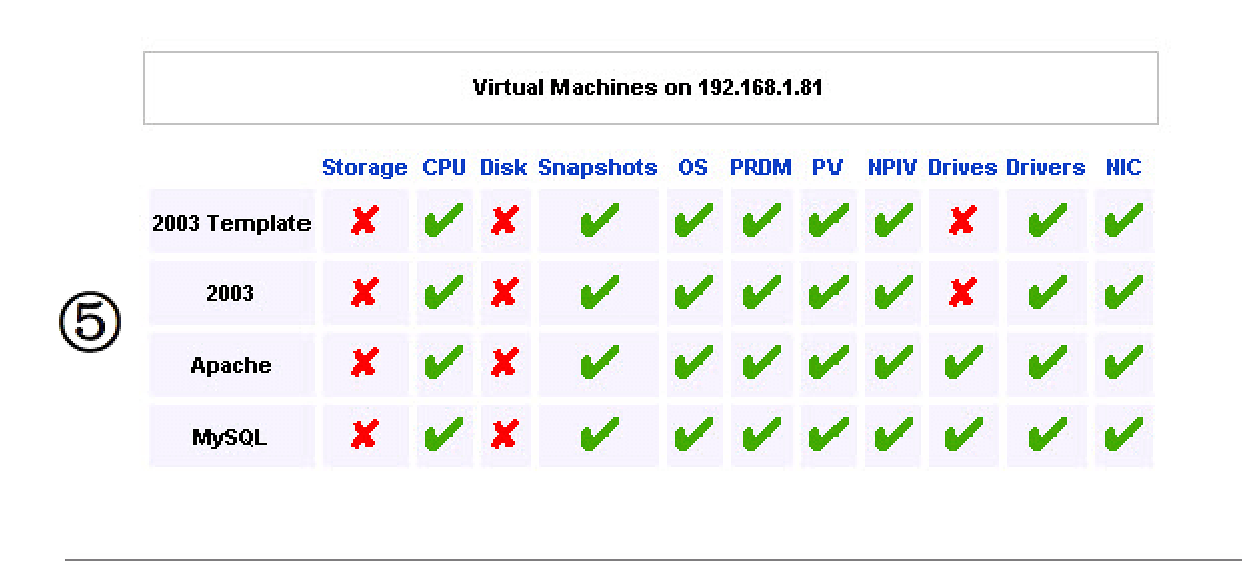
4. 此时会开始生成报表。



▲ 键入账号口令及 IP

▲ 开始生成报表

5. 生成后的报表会在浏览器中弹出。



⑤

Virtual Machines on 192.168.1.81

	Storage	CPU	Disk	Snapshots	OS	PRDM	PV	NPIV	Drives	Drivers	NIC
2003 Template	✘	✔	✘	✔	✔	✔	✔	✔	✘	✔	✔
2003	✘	✔	✘	✔	✔	✔	✔	✔	✘	✔	✔
Apache	✘	✔	✘	✔	✔	✔	✔	✔	✔	✔	✔
MySQL	✘	✔	✘	✔	✔	✔	✔	✔	✔	✔	✔

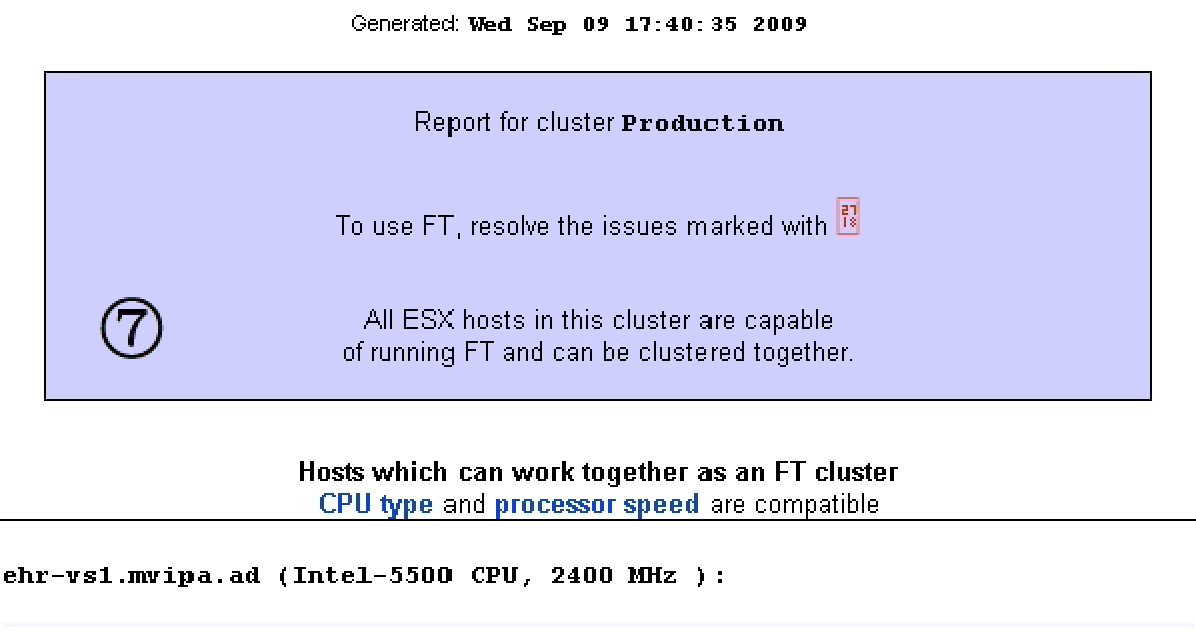
▲ 弹出了报表

6. 在浏览器中，我们可以看到生成的报表。会给出系统的 CPU 以及是否支持 FT。下图中为不支持。



▲ 图中为不支持

7. 如果你的 CPU 和系统支持 FT，就会给出完整的报告。



▲ 若系统支持 FT，会有完整的报告

18.3.2 VMware vSphere 的先决条件

当你的 CPU 准备好，VMware 的版本也支持 FT 功能后，接下来就是准备 vSphere 的 Cluster 配置及 VM 配置了。我们就来看看运行 FT 的一些先决条件。

1. FT 的硬件条件

这是 FT 实行的遍历列表，一定要确定硬件能支持这里的列表才可行。

▶ FT 硬件需求遍历列表

- CPU 必须是同一家族的产品，不支持混用。
- 要激活 FT 的 VM 只能有一个 vCPU。
- 在 BIOS 中一定要先将虚拟化的命令打开。
- FT 的主机必须在 VMware 激活 HA 的 Cluster 中。
- 一定要使用独立存储（FC SAN、NFS、iSCSI）。
- 一定要有一个 Service Console 和 VMotion 在一个 vSwitch，一个 FT Loggin 的 VMkernel 在另一个 vSwitch。
- 必须使用 GB 级的网卡，最好是 10GB 的网卡。
- 使用 Jumbo Frame。
- 将单点失误降到最低，如 NIC Team、多个外部存储、多个网络交换机、使用多路径来访问存储设备。
- 运行的 ESX 主机必须运行完全一样的 ESX 版本及升级程序。

2. 运行 FT 的 VM 条件

在知道了硬件的需求后，在 VM 上也有一些特殊的规则要遵守，才能常规激活 VM 上的 FT 功能。

▶ 在 VM 上的需求遍历

- 正本和副本的 VM 一定要在同一个激活 HA 的 Cluster 中。
- 正本和副本的 VM 不可以在同一台 ESX 主机上运行，这样就失去了 FT 的意义。
- VM 的所有文件包括 VMX 和 VMDK 都必须存在独立存储上。
- VMDK 的格式不可以是 Thin Provisioned，必须是 Thick，如果是 Thin 的话，必须先转换成 Thick，因此可能会需要额外的磁盘空间。

- 当你将 VM 的 FT 激活时，这个 VM 的磁盘格式会被自动转换。
- 虚拟 RDM 模式支持，但物理的 RDM 模式不支持。

18.4 正规实践 FT

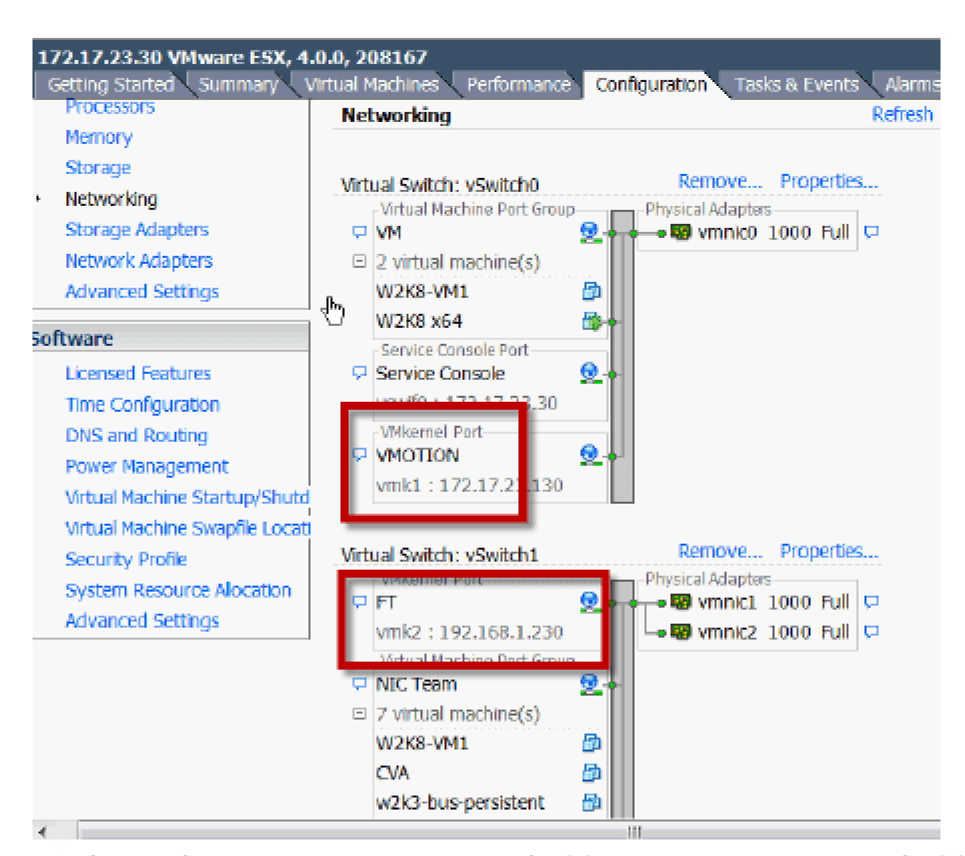
在理解了 FT 的原理及条件之后，剩下实践 FT 就相当简单了。实践 FT 最重要的就是配置 FT Logging 的 VMkernel，只要这部分落实，再加上硬件的支持以及集群的配置落实，就可以使用了。

18.4.1 生产环境的遍历任务

由于 FT 无法运作在 VMware Workstation 所搭建的实验环境（CPU 制约），因此我们只能在生产环境上实践 FT，我们就先看看这个生产环境是否能运行 FT。

1. 遍历网络配置

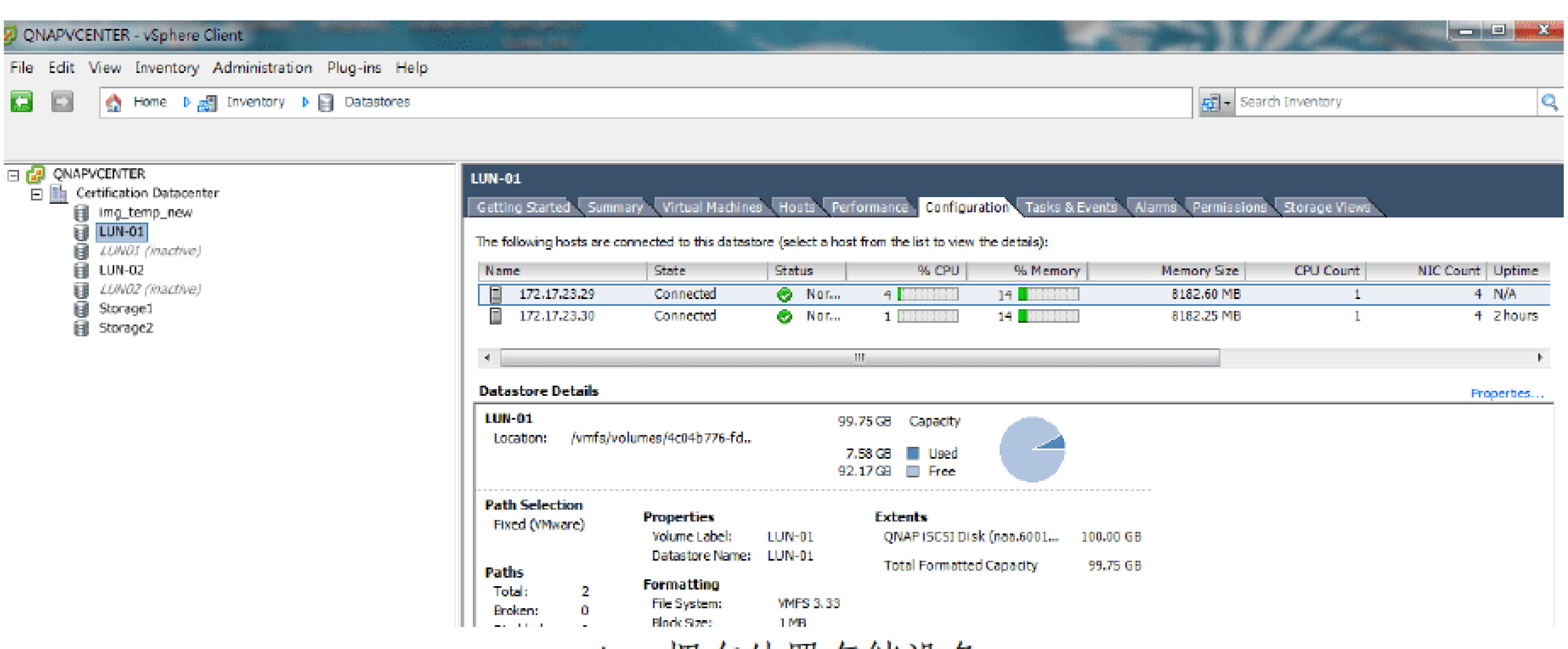
在 FT 环境中最重要的就是要有 VMotion 和 FT Logging 两个 VMkernel。我们在生产环境中，已经有一个 VMotion 的 VMkernel（名称为 VMOTION）和 FT 的 VMkernel 了（名称为 FT）。



▲ 要有两个 VMkernel，一个是 VMotion，一个是 FT

2. 遍历存储配置

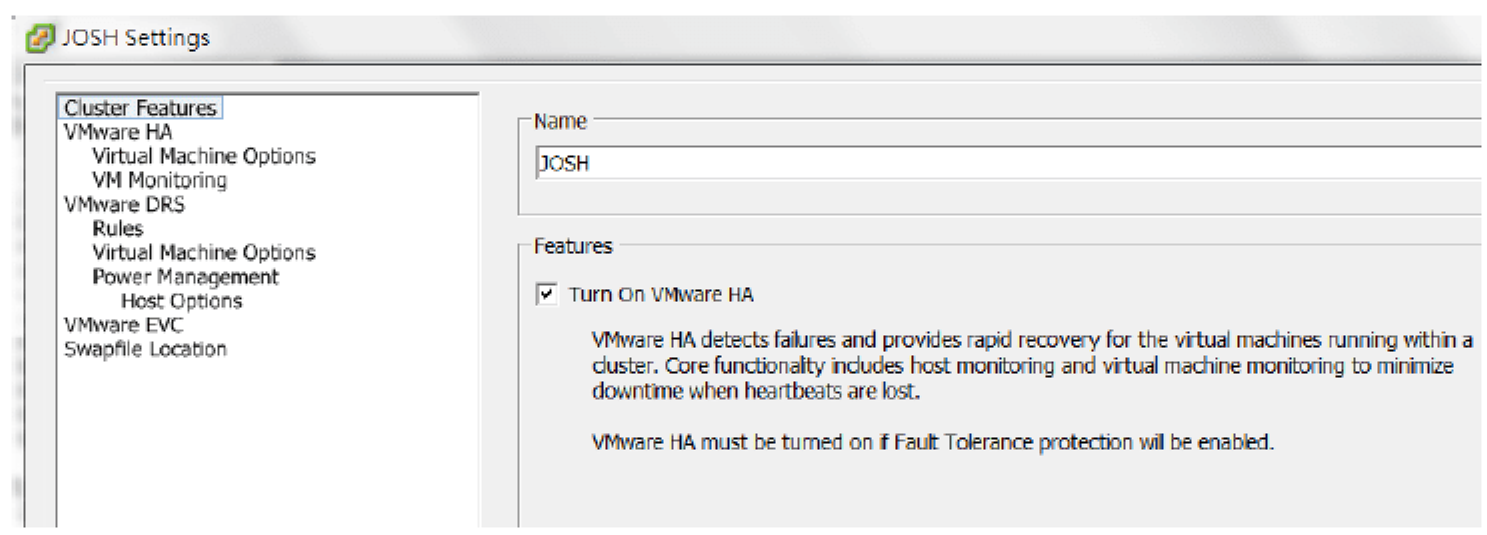
在生产环境中，我们已经有一个 LUN0 和 LUN1 的 iSCSI 外部存储设备，并且可以被两台进行 FT 的 ESX 主机访问。



▲ 拥有外置存储设备

3. 遍历 Cluster 配置

生产环境的 Cluster 只要有激活 DRS 和 HA 即可，并且在激活后不可以有任何的错误消息。



▲ Cluster 要激活 DRS 和 HA

4. 遍历 CPU

你可以运行 SiteSurvey 或是直接在 vCenter 下检视 CPU 是否匹配 FT 的要求，我们的生产环境当然是匹配要求的。



▲ 生产环境的的 CPU 是匹配 FT 的

18.4.2 进行 FT 实践

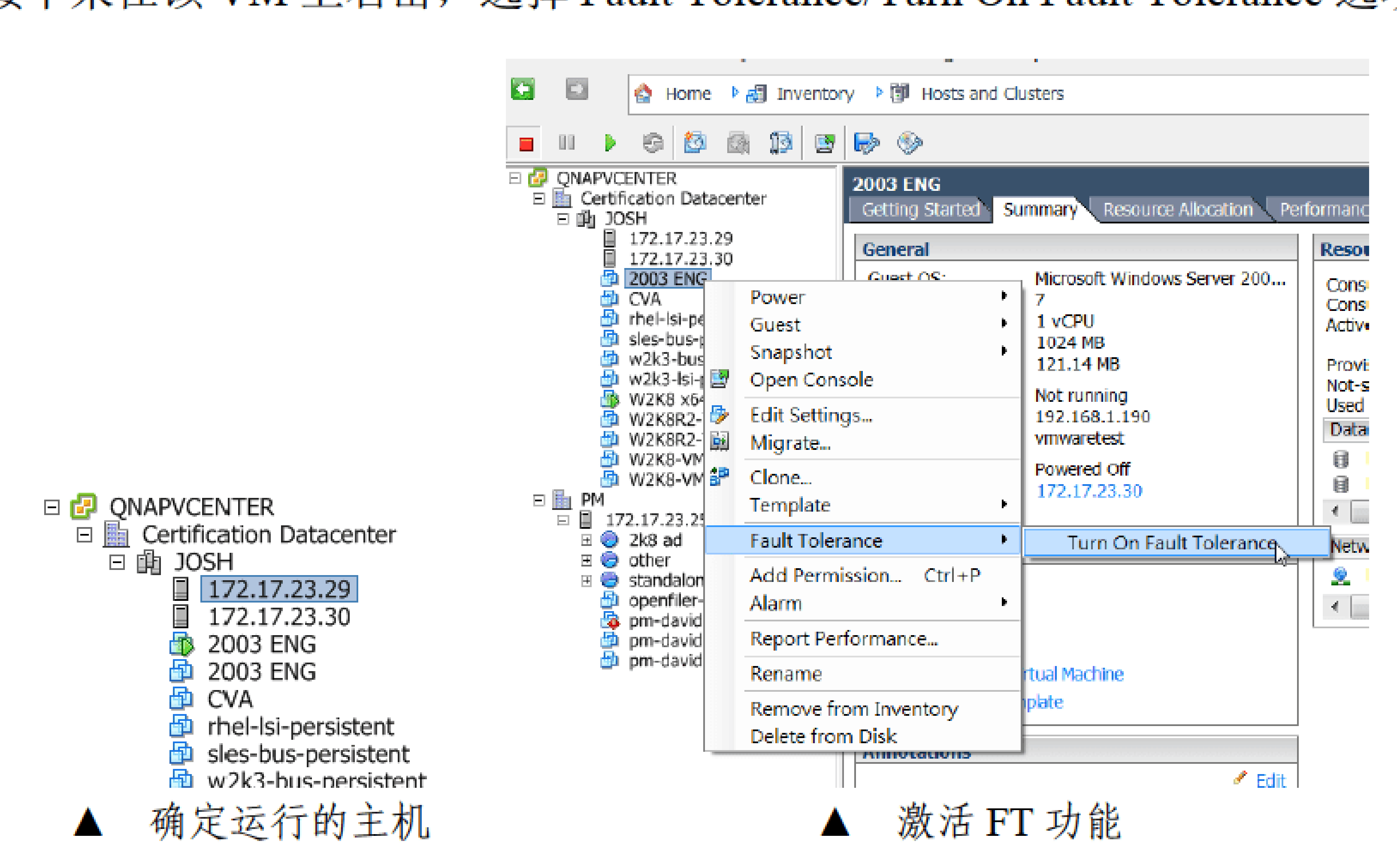
终于等到令人兴奋的一刻了，接下来我们就进行真枪实弹的 FT 实践，热点当然就是找一台已经引导的 VM 来进行 FT 的任务，下面就是完整的步骤。

1. 对 Windows 2003 的 VM 进行 FT

接下来就是针对 Windows 2003 的 VM 进行的 FT 完整步骤。

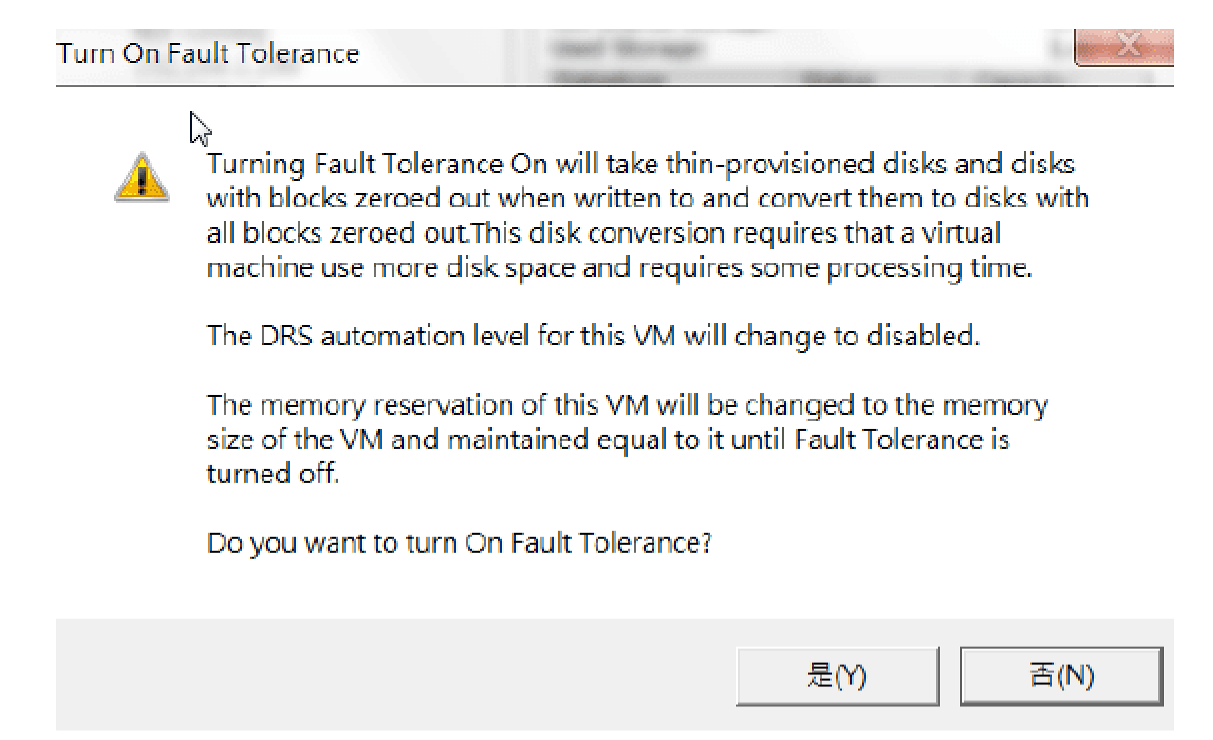
▶ 对 Windows 2003 的 VM 进行 FT

1. 进入 vCenter 中，并且确定当前运行的 Windows 2003 的 VM 的位置。当前该 VM 的名称为 2003 ENG，存放的位置是在 172.17.23.29 这台 ESX 主机上。
2. 接下来在该 VM 上右击，选择 Fault Tolerance/Turn On Fault Tolerance 选项。



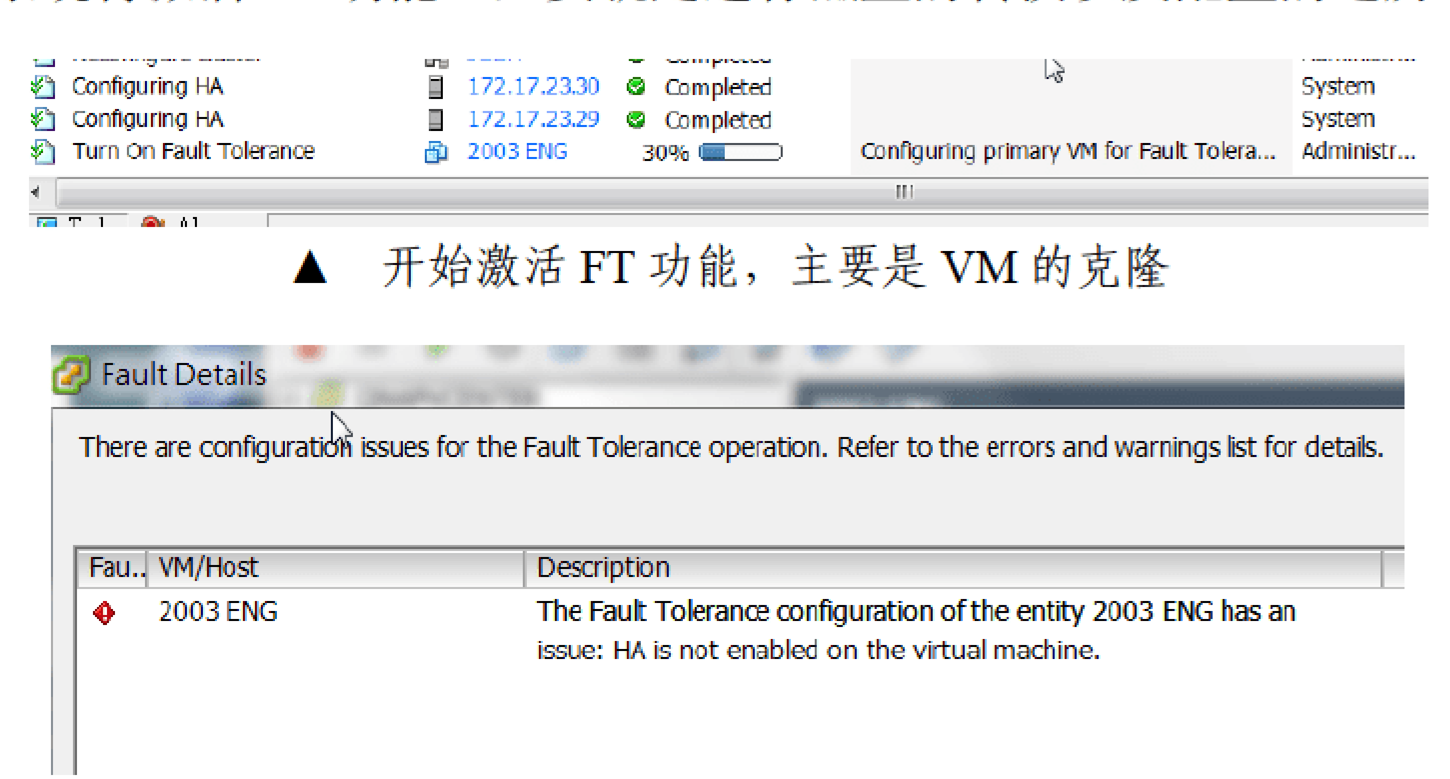
▲ 确定运行的主机　　▲ 激活 FT 功能

3. 此时会弹出一个报警画面，意味着会将原来 Thin Provisioned 的磁盘变成 Thick Provisioned。单击“是（Y）”按钮。



▲ 这里是转换磁盘的报警

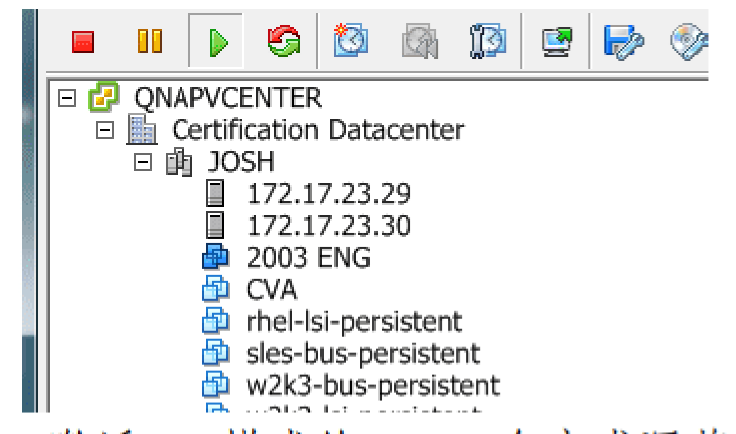
4. 此时系统将激活 FT 功能，主要就是进行磁盘的转换以及配置的遍历。



▲ 开始激活 FT 功能，主要是 VM 的克隆

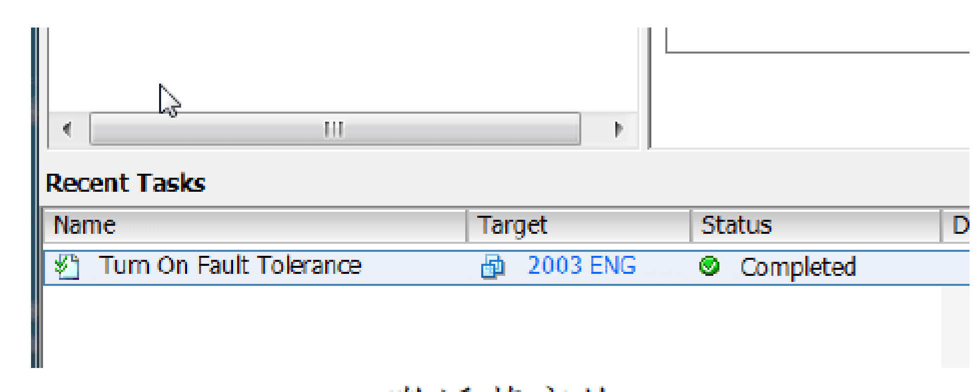
▲ 如果你的环境不匹配 FT 条件，就会弹出各种不同的错误消息

5. 当激活 FT 时，你会发现 VM 的图示颜色变为深色，意味着这个 VM 已经激活成为 FT 模式。

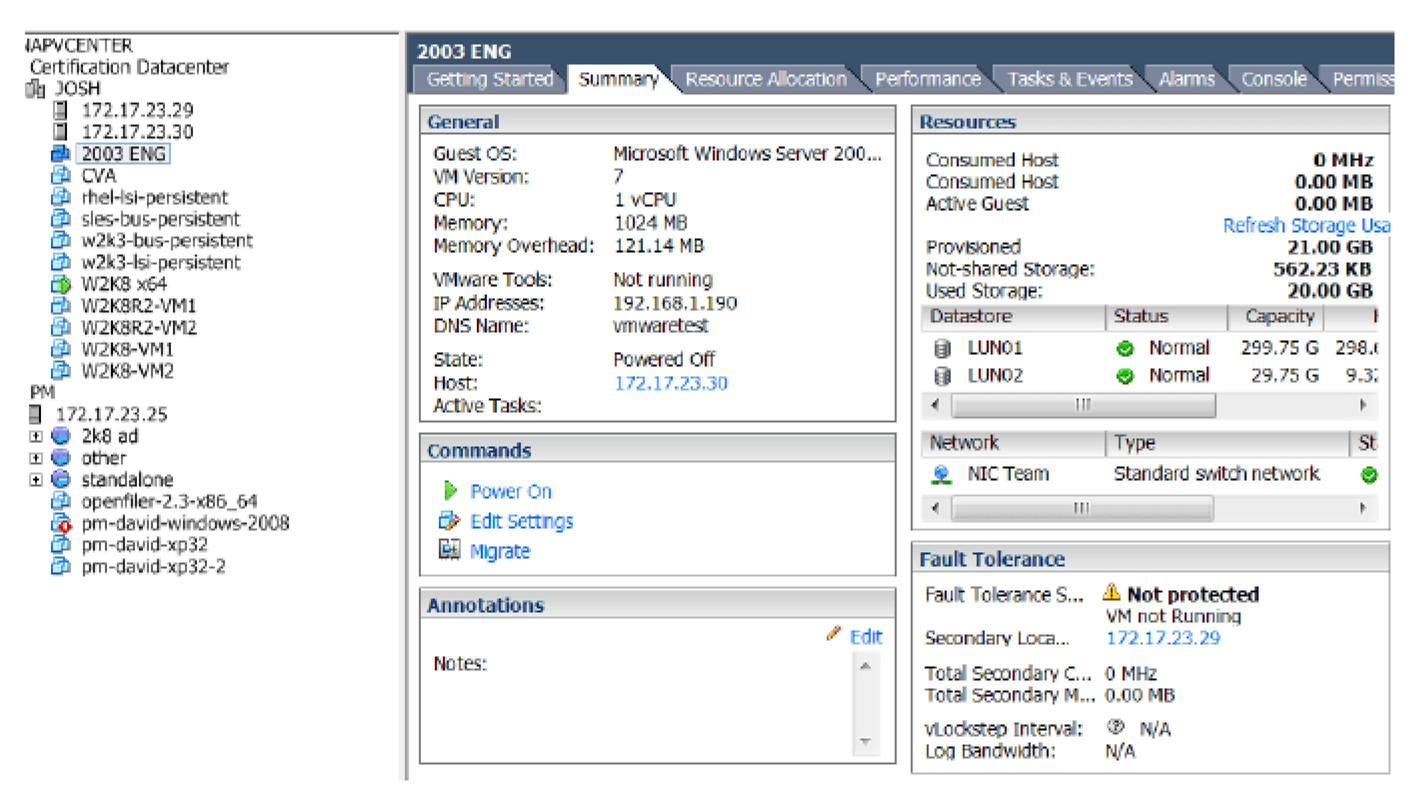


▲ 激活 FT 模式的 VM，会变成深蓝色

6. 此时系统会开始激活 FT 功能，但还没有准备好。此时如果在该 VM 上观察，会发现 Fault Tolerance 的部分还没有激活，会显示 Not Protected 字样。但我们也可以看到 VM 的位置会在 172.17.23.30。

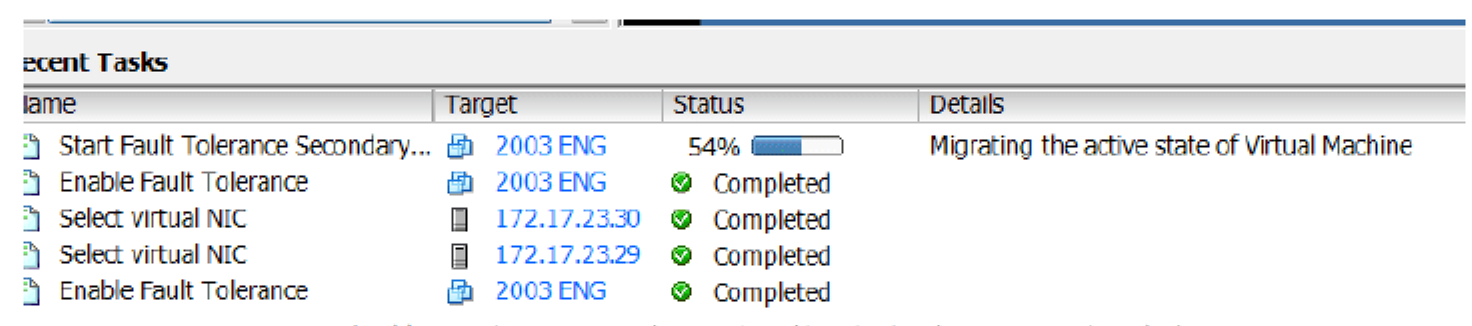


▲ 激活落实的 VM



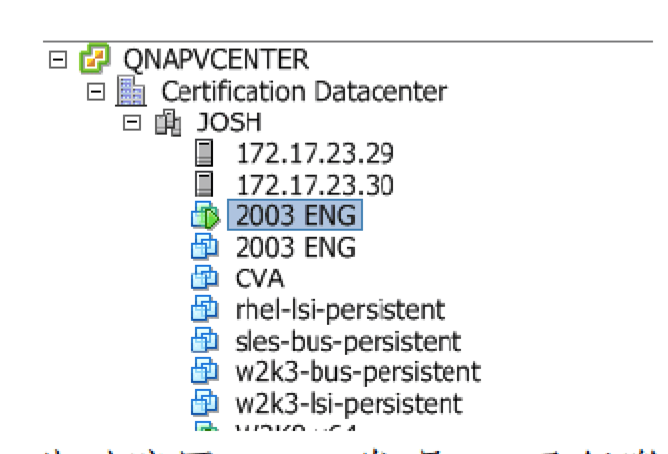
▲ 此时还没有完全克隆好，因此在该 VM 上会弹出 Not Protect 字样

7. 接下来落实 FT 激活。将正本 VM 的状态移植到副本 VM 上。



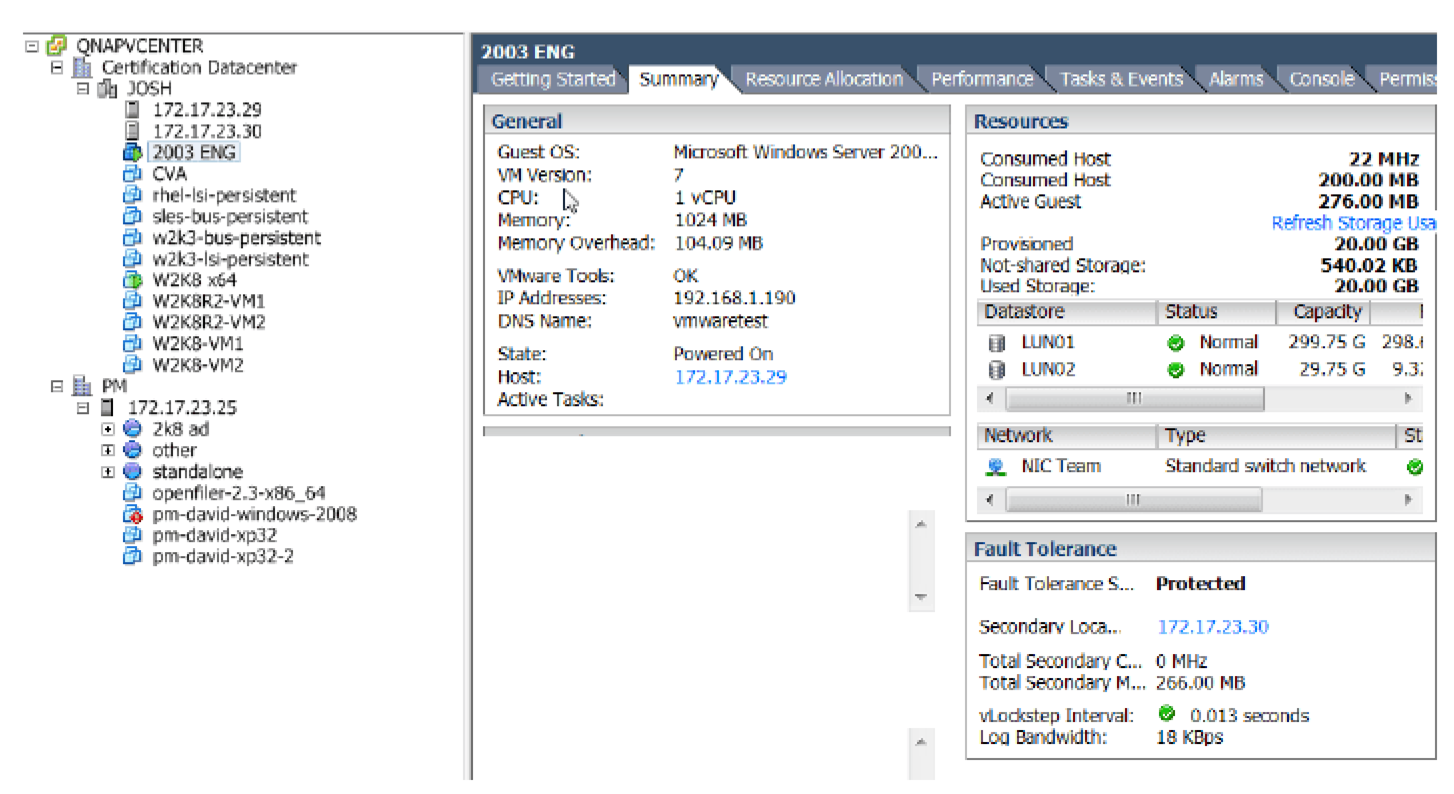
▲ 在第二台 ESX 上已经激活这个 VM 的副本了

8. 当副本的 VM 激活后，我们可以看到 Fault Tolerance 的状态也已经激活了（Protected），并且可以看到副本（Secondary）的位置是在 172.17.23.30 上，当前副本的 CPU 使用是 0MHz、内存是 266MB，而两者之间的时间差（vLockstep interval）是 0.013 秒，表项 LOG 的带宽是 18KBps。



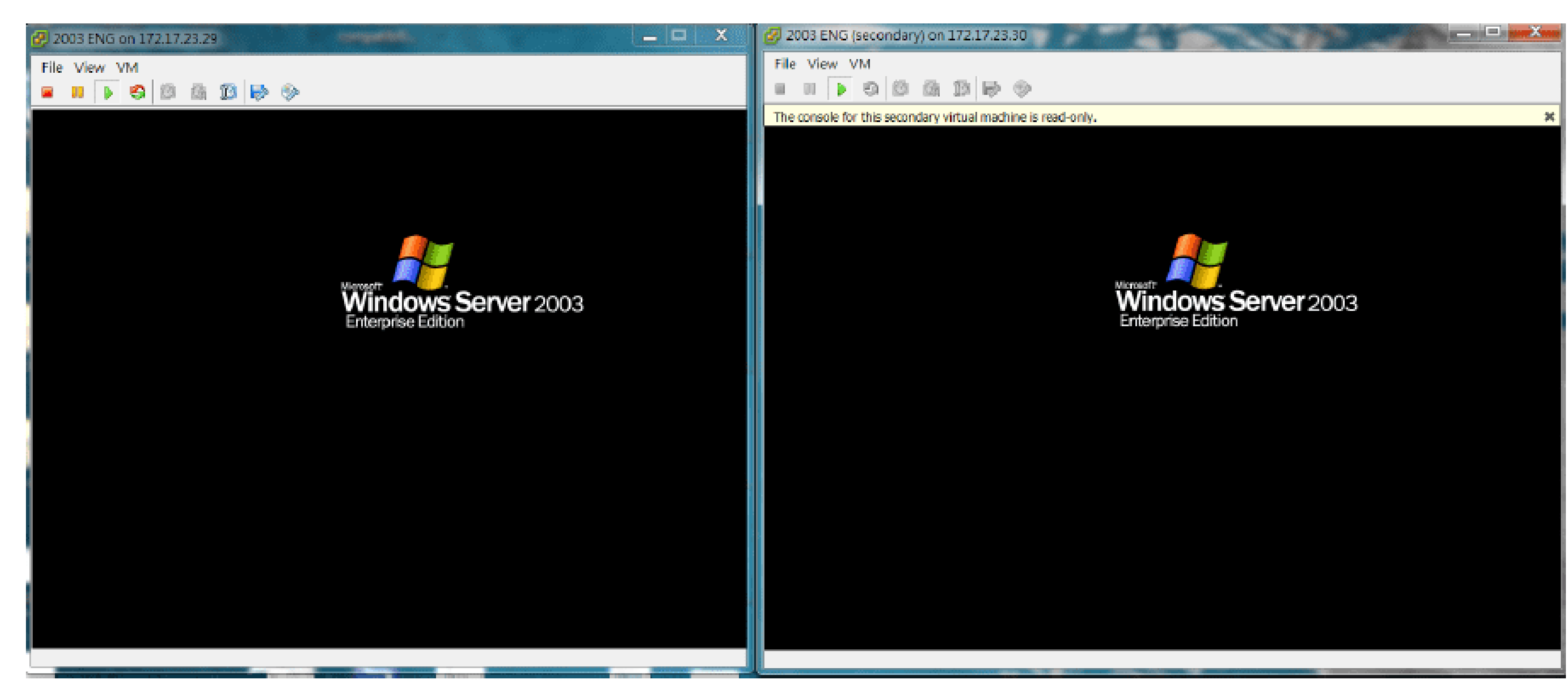
▲ 此时遍历 VM，发现 FT 已经激活了

9. 此时我们到 172.17.23.30 上看，会发现有一台 2003 ENG（secondary），就是副本的 VM。



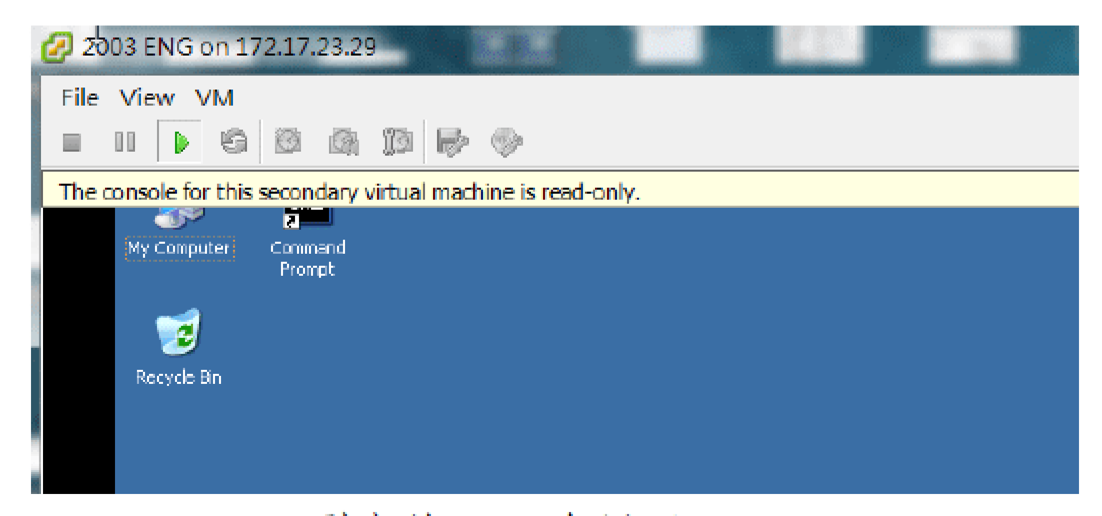
▲ 这就是这个 VM 在另一台机器上的副本

10. 接下来我们可以进入两个 VM 的主控台，将两个 VM 的主控台打开，进入正本的 VM 来操作，你会发现两边的动作完全一样，你在正本操作的所有动作，在副本上几乎没有延迟完全一样。



▲ 完全一样的操作，好像影子一样

11. 但要注意的是，副本的 VM 主控台会弹出 Read-only 的字眼，意味着你无法操作副本的 VM。



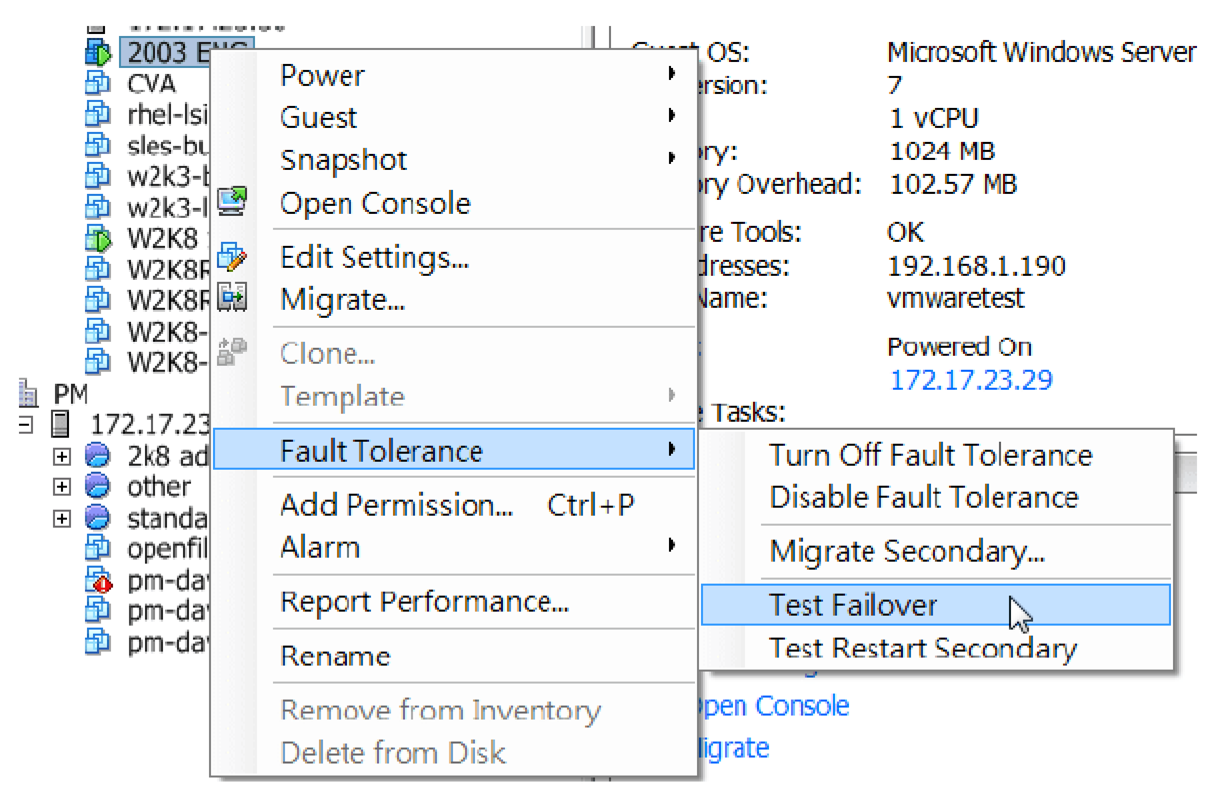
▲ 副本的 VM 会显示 Read-only

2. 测试 FT 是否真正启用

vSphere 提供了一个测试 FT 的功能，此时会将正本 VM 所在的 ESX 主机暂时停止一下。我们可以利用这个功能来测试副本的 VM 是否启用。

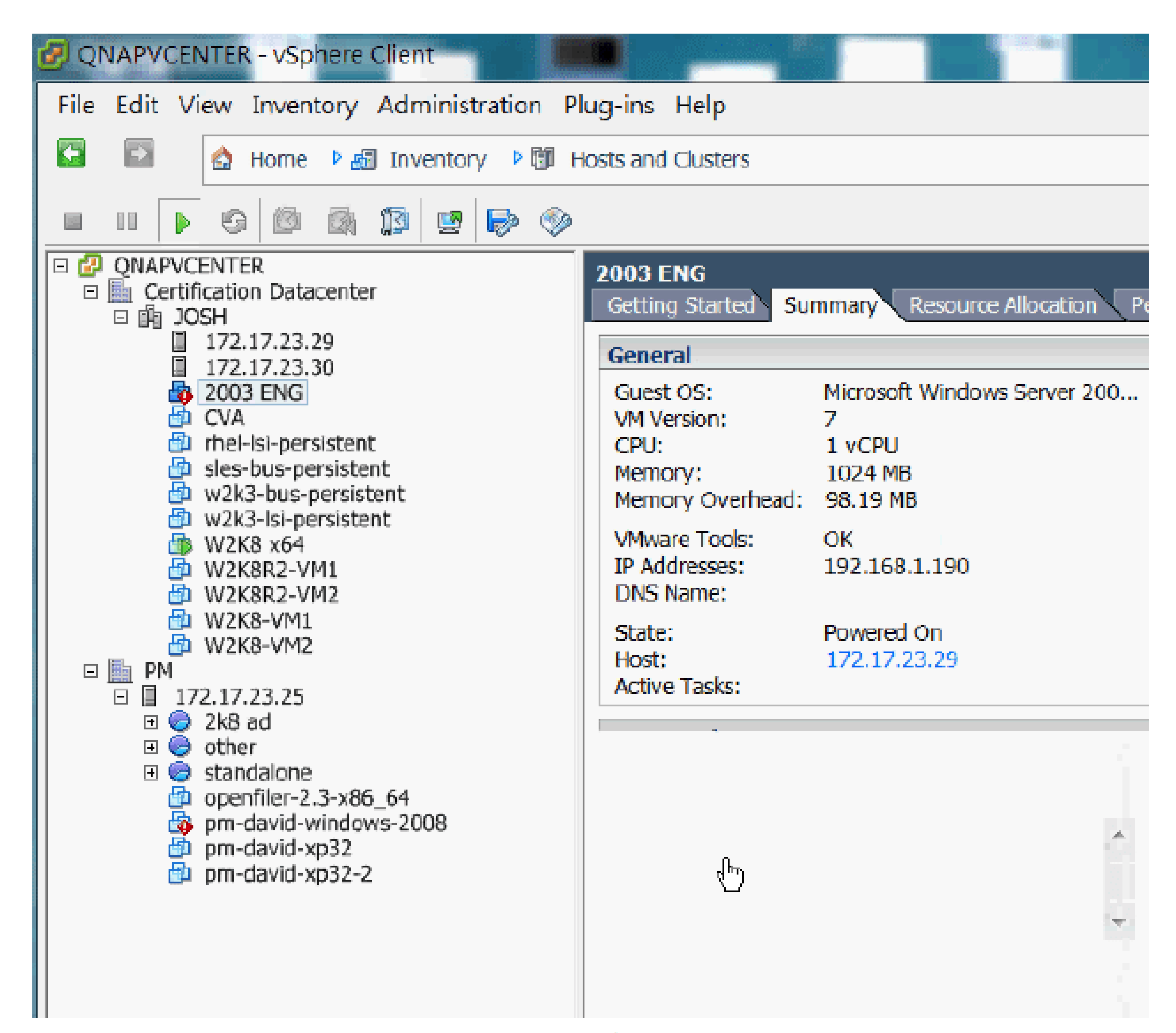
▶ 测试 FT 是否启用

1. 在 FT 激活的情况下，在该 VM 正本右击，选择 Fault Tolerance/Test Failover 选项。



▲ 激活测试 FT 的功能

2. 此时系统会将正本的 VM 停用，如下图所示。



▲ 正本被停用了

但副本的 VM 会立即启用，并且成为正本的 VM。如果此时有任何问题，系统会提出报警，一般来说，如果你的 VM 能成功激活，测试大部分会通过，除非副本 VM 所在的 ESX 的资源已经不够。

结 语

FT 的原理虽然复杂，但实践却很简单，FT 的出现，让 IT 管理员梦想已久的完全不蓝屏环境终于成真。有了 FT 功能，不但可以省下大笔双机热备的成本，更可以在已经落实的 IT 架构上创建永不蓝屏的环境，这也是为什么 vSphere 一推出之后，大家关注的焦点都集中在 FT 上，如果你的企业环境真的不允许任何时刻的蓝屏，就快点装上 FT 吧！

读书笔记

第 19 章 在 vSphere 下直接访问硬件

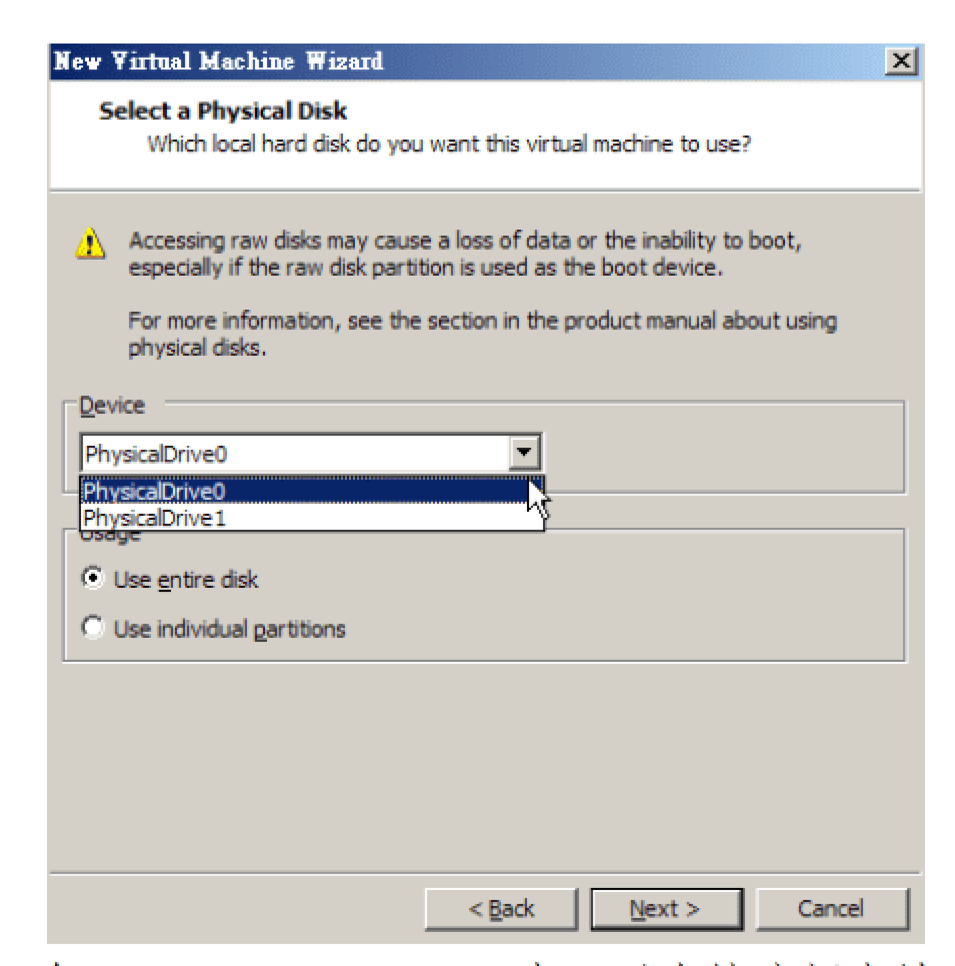
关键词：

- 什么是 RDM 模式
- RDM 模式的使用时机
- RDM 模式使用时的系统需求
- 在 ESX 中具体操作 RDM 模式
- 理解直接硬件访问是什么
- 理解直接硬件访问的模式
- CPU 和芯片组的配合
- 在 vSphere 中操作直接访问硬件

vSphere 提供了一个 Hypervisor 来管理硬件，并且将上层所有操作系统的命令全部利用二进制翻译（Binary Translation）来转换所有的命令。还好有了 CPU 的 VT—x 命令，让这个动作只开销掉整体资源的 3%～10%。然而在许多情况下（如兼容性或是效能），VM 必须直接访问某一个硬件资源，然而在虚拟化之后的所有硬件都无法让 VM 直接访问，难道这时就不能用 VM 了吗？vSphere 提供了 VMDirectPath I/O pass—through 以及 RDM 模式，就是为了解决这个问题，我们就来看看吧。

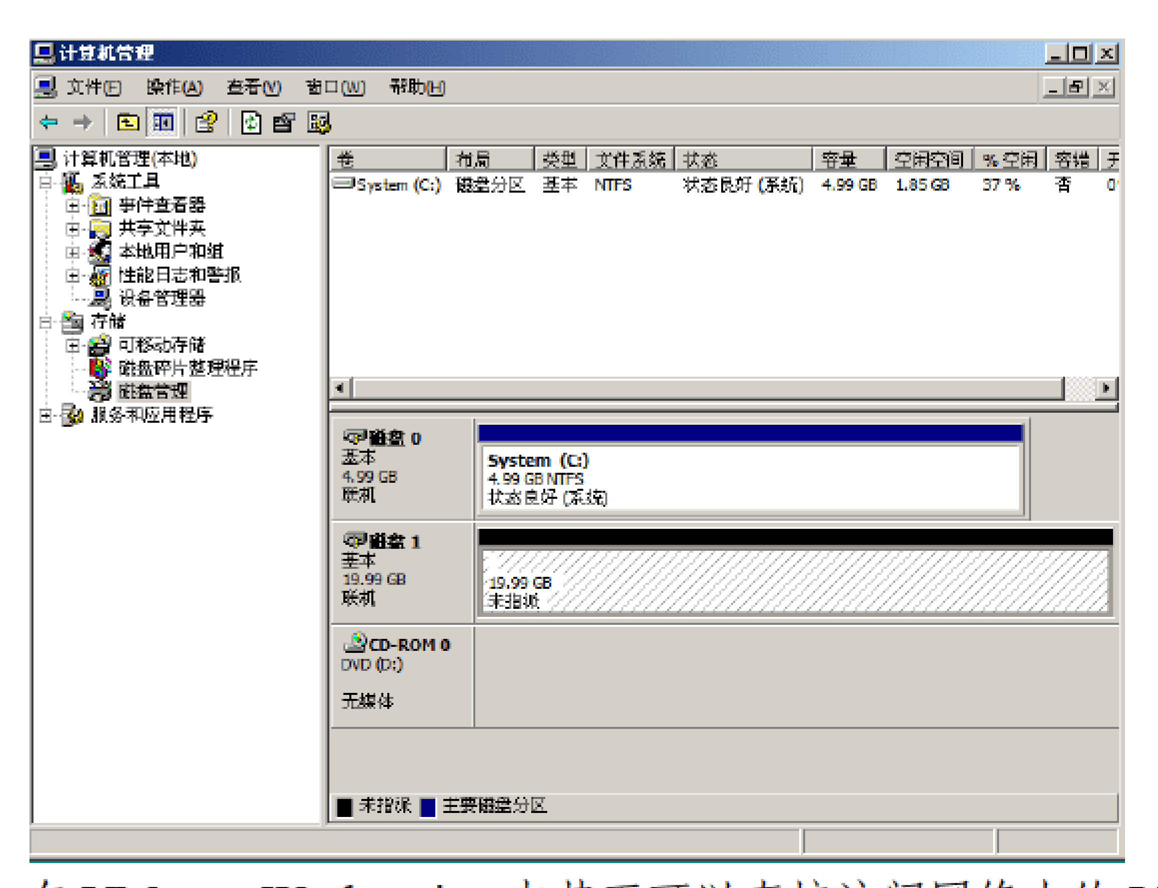
19.1 RDM 模式

常常使用 VMware Workstation 的读者们，一定很清楚 Workstation 版本中对于本地硬盘的直接访问功能。你可以直接让 VM 掌管一个真实的硬件或分割区，直接将操作系统安装在这个硬盘中。这么做的好处除了可以让这个硬盘拿到其他的 VM 引导之外，更可以保证 VM 运行时，不会因为使用 VMDK 需要 CPU 来转换，导致效能的低落。在 VMware Workstation 中，只要 Windows 能找到这个硬盘，不管是 SATA、IDE、SCSI 或是 USB 硬盘，都可以拿来给 VM 直接使用。



▲ 在 VMware Workstation 中可以直接访问本地硬盘

这种做法的确给 VM 很大的弹性。举例来说，如果有一个 USB 随身盘，我们使用直接硬盘访问的方式将操作系统安装在这个 USB 随身盘中，我们可以很容易地将这个 USB 随身盘拿到其他有安装 VMware Workstation 的机器上引导，同时也能确定效能的优势。那么这么好的功能，在更高级的 vSphere 版本上也能做到吗？



▲ 在 VMware Workstation 中甚至可以直接访问网络上的 LUN

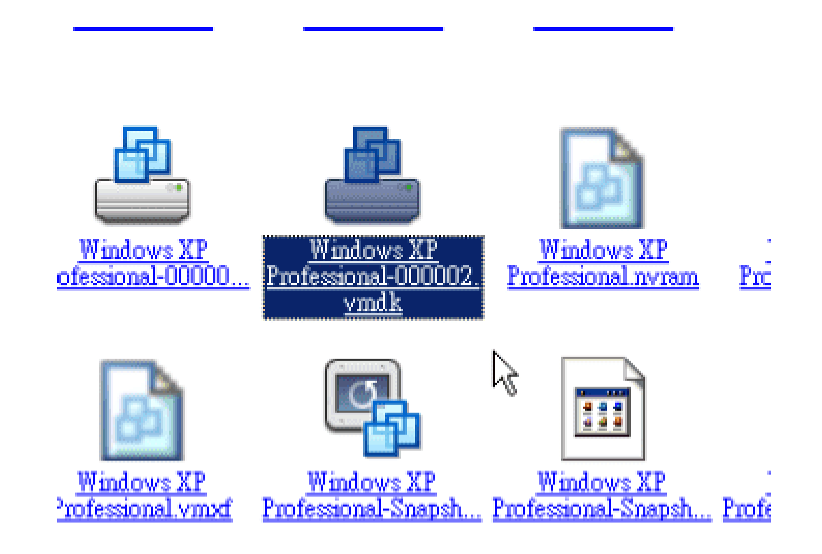
19.1.1 vSphere 中的 RDM 模式是什么

在 vSphere 的 ESX/ESXi 主机上，系统并不允许 VM 直接访问其上的独立磁盘，这是对安全

及企业应用的兼容性所考虑；但不能访问本地的磁盘，不代表 VM 不能访问网络上的逻辑磁盘单位。而这种访问网络 LUN 设备的功能，在 vSphere 中称为原生设备映设（Raw Device Mappings），简称为 RDM。

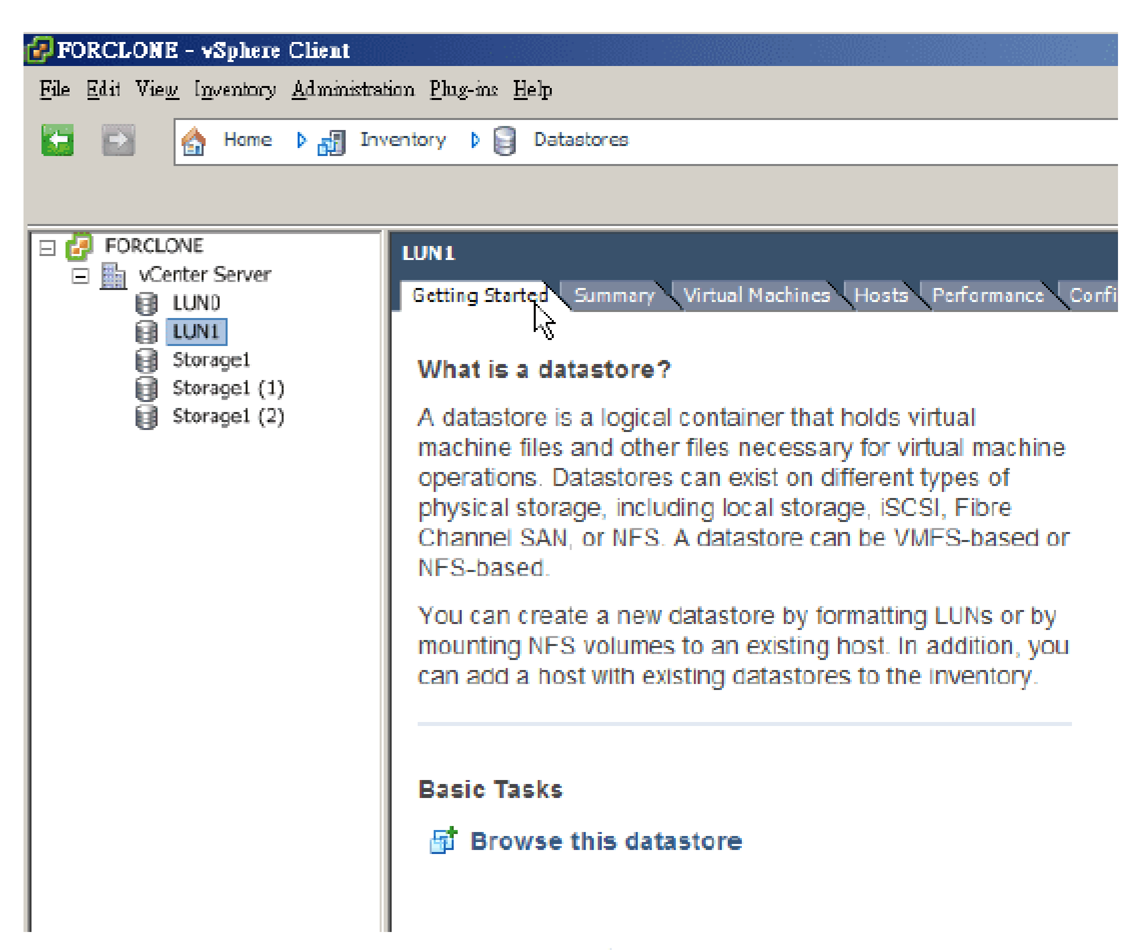
1. 直接访问 LUN

RDM 的模式十分简单，在创建 VM 时，原来的模式是在一个 VMFS 的文件系统中划出一个 VMDK 文件作为虚拟硬盘。VM 上操作系统所针对硬盘的 I/O 可读都会被 Hypervisor 转换，最后变成对 VMDK 文件的可读，因此在时间上会有一层转换的延迟，对于十分讲究磁盘效能的应用来说（如数据库），VMDK 虚拟硬盘的方式在海量数据 I/O 时会产生严重的效能瓶颈。



▲ 一般来说 VMDK 就是一个虚拟硬盘，任何动作都要经过转换

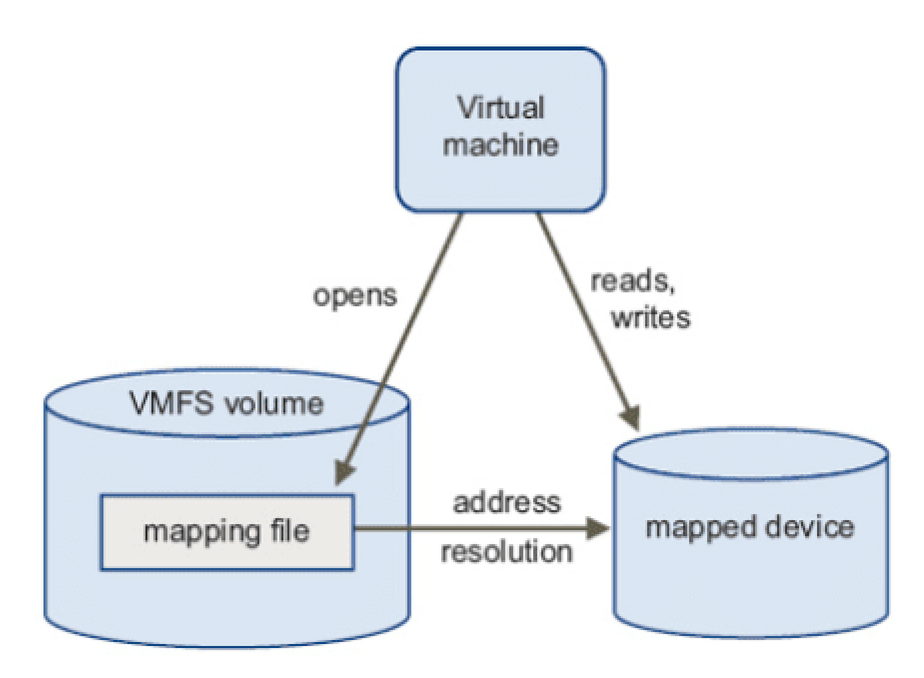
RDM 模式让 VM 上的操作系统直接访问网络上的 LUN，不管是 FC SAN 或是 iSCSI 的 LUN。这时访问的方式不再将 LUN 视为 Datastore，而是一个“独立的磁盘”。这个独立的磁盘事实上就是存储设备上的一个 LUN，直接在 VM 上也视为一个硬盘。



▲ 一般 LUN 会被视为 Datastore

在 VMDK 模式时，LUN 是被 ESX/ESXi 挂成存储，并且以 Datastore 的方式来存放，这个 LUN 会被格式化为 VMFS 格式，VM 的虚拟硬盘会以 VMDK 的文件格式存放在这个已经成为 VMFS 格式的 Datastore 的 LUN 中。

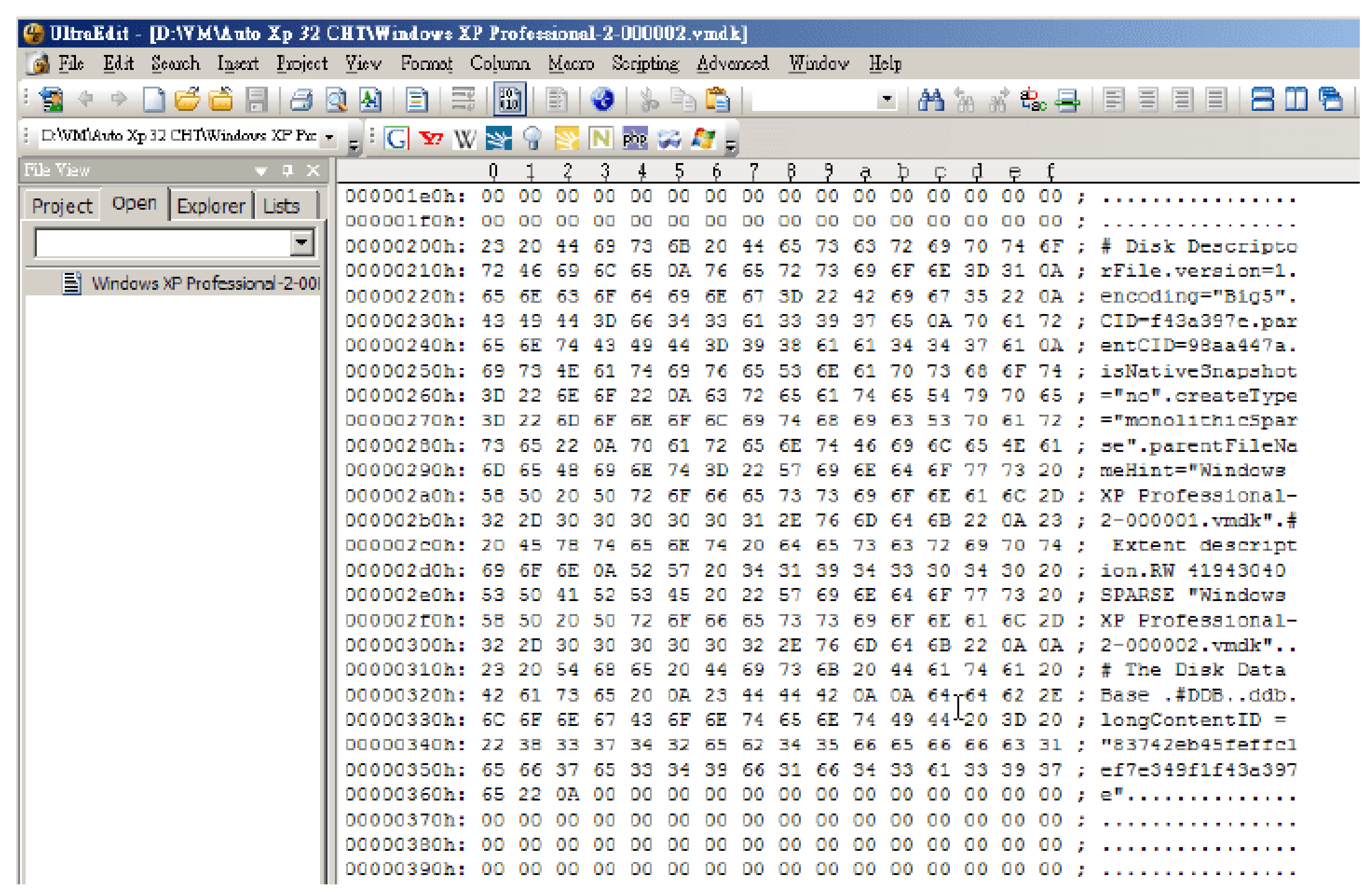
但是在 RDM 模式时，LUN 是被 VM 视为一个独立硬盘的，这个 LUN 可以是各种文件格式，如 NTFS、Ext3/4、FAT32 等，视总控这个 LUN 的操作系统来决定。VM 可以用 Bit by Bit 可写硬盘的方式直接可写这个 LUN，而不需要通过 Hypervisor 的翻译，如此一来，不但效能可以大幅提高，更添加了 VM 使用的弹性。



▲ RDM 模式时，LUN 就是 LUN，不管是 VM 或是物理机都是 LUN

2. 近距离观察 RDM 模式的 VMDK

在非 RDM 模式时，VMDK 的文件就是虚拟硬盘档，其中也含有 Meta Data，但这个文件主要还是一个二进制文件，用来存放真正的硬盘数据，如图所示。



▲ 在虚拟硬盘时，VMDK 是一个二进制文件

在 RDM 模式时，虽然系统仍然会创建一个 VMDK，但此时的 VMDK 已经不再是虚拟硬盘，而是一个对应的文本档。RDM 模式虽然是直接访问，对于 Raw Device 的访问，VM 还是会通过一个 Meta File 来访问物理硬盘。而这个 Meta File 就是这个 VMDK，在 VMware 中称为 Disk DescriptorFile。在 RDM 模式时，VMDK 就是纯文本档了。其中记载着这个 VM 对应到物理硬盘的数据。我们可以看到文件的内容。



▲ 在 RDM 模式时，变成一个纯文本档

19.1.2 RDM 的种类

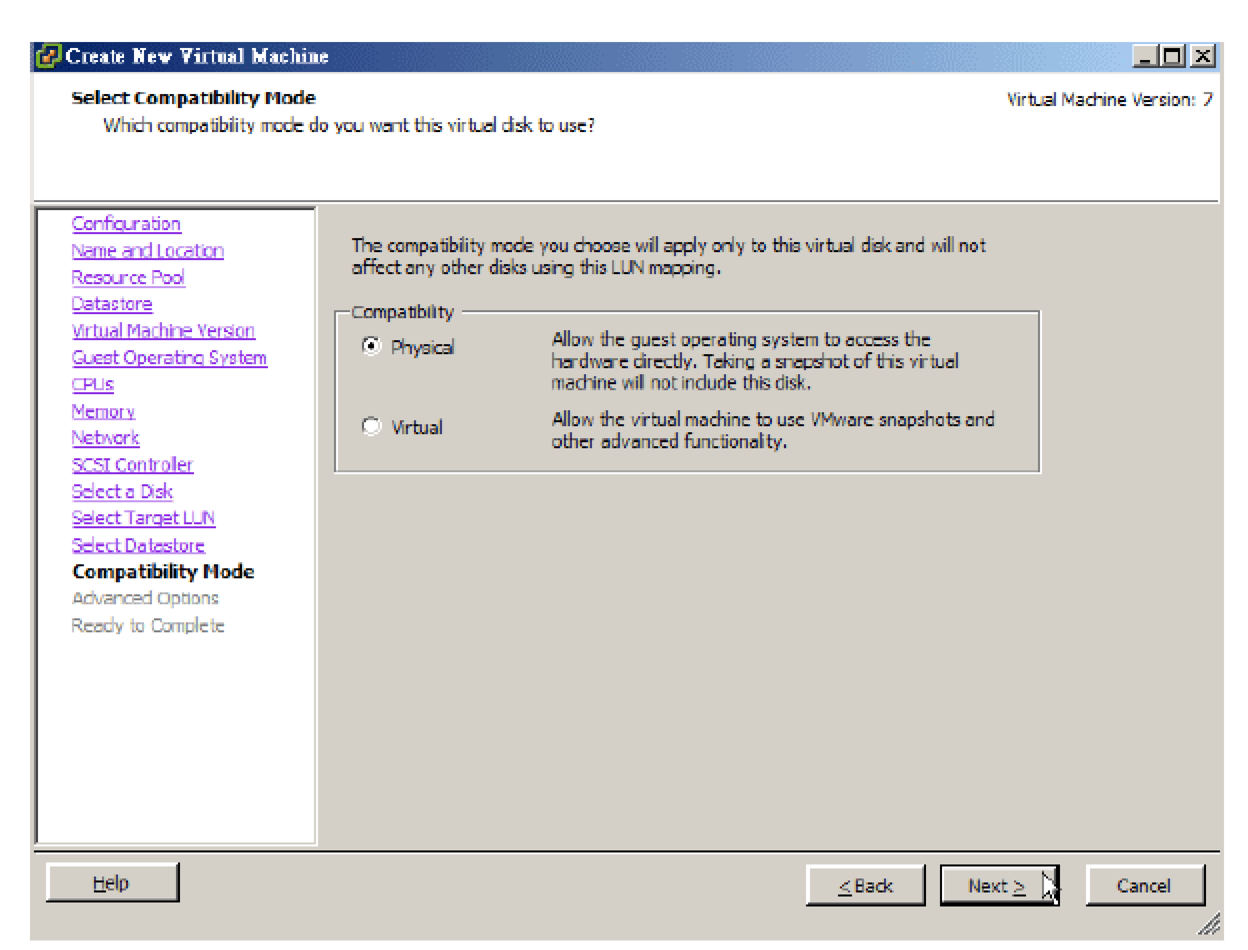
在 vSphere 中，RDM 共有两种模式，我们就来看看。

1. Physical compatibility mode

第一种称为物理兼容模式，在这种模式中，除了针对 LUN 上层操作的命令被隐藏起来之外，VMkernel 将所有的 SCSI 命令直接提交到 LUN 设备上，如此一来，VMkernel 才能独立该 LUN 给 VM 使用。在这种模式，SAN 管理界面程序是置放在 VM 中的，或 VM 中有其他 SCSI 的目标软件才能运作。物理模式最大的优点是速度快，但 vSphere 上的大部分的虚拟操作则无法使用，如快照模式。

2. Virtual compatibility mode

在虚拟兼容模式中，真实的硬件特征依旧被隐藏，所有 VMFS 特性则仍然保留，如 file locking 与快照等机制。虚拟兼容模式的优点就是能继续使用 VMware 的功能，但效能就没有物理兼容模式快了。



▲ 在创建 RDM 模式时可以选择这两种

19.1.3 RDM 模式的使用时机

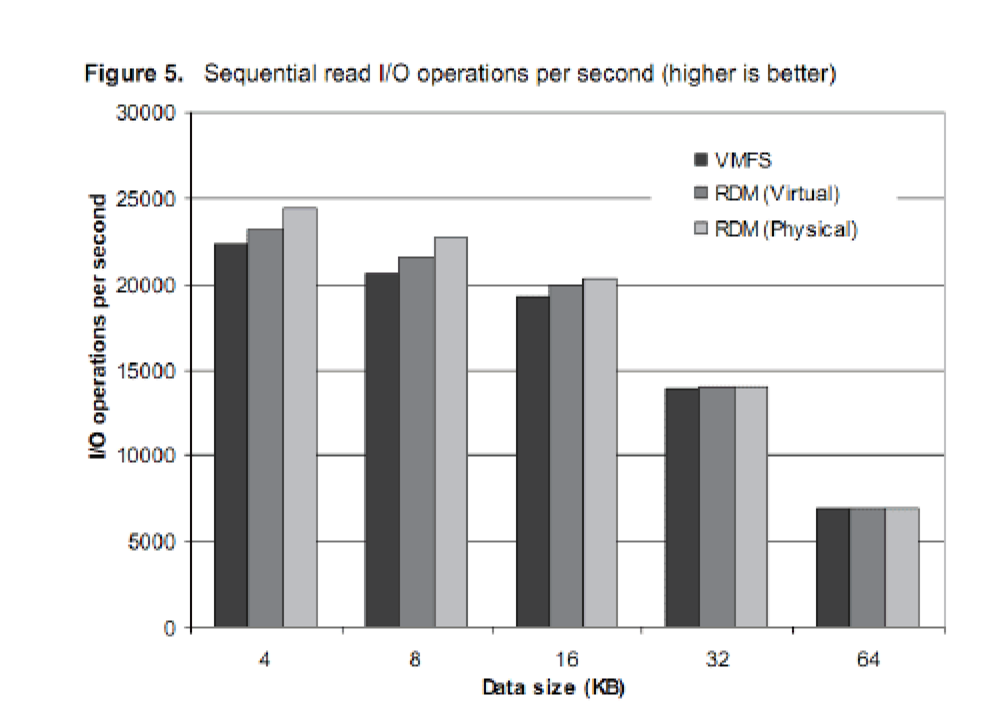
RDM 的使用时机通常就是效能和兼容性，另外在 P2V 的过程中，如果使用了 RDM，更可以加快 P2V 的速度，下面就分别来看看。

1. 需要效能的时机

VMware 本身针对 RDM 进行效能测试，发现 RDM 模式在单位传输、寻轨时间及 CPU 使用率上，RDM 都领先同环境的 VMDK 模式。有兴趣的读者可以到下面的网址来检视完整的测试环境。www.vmware.com/files/pdf/vmfs_rdm_perf.pdf。

2. 方便 P2V 的转换

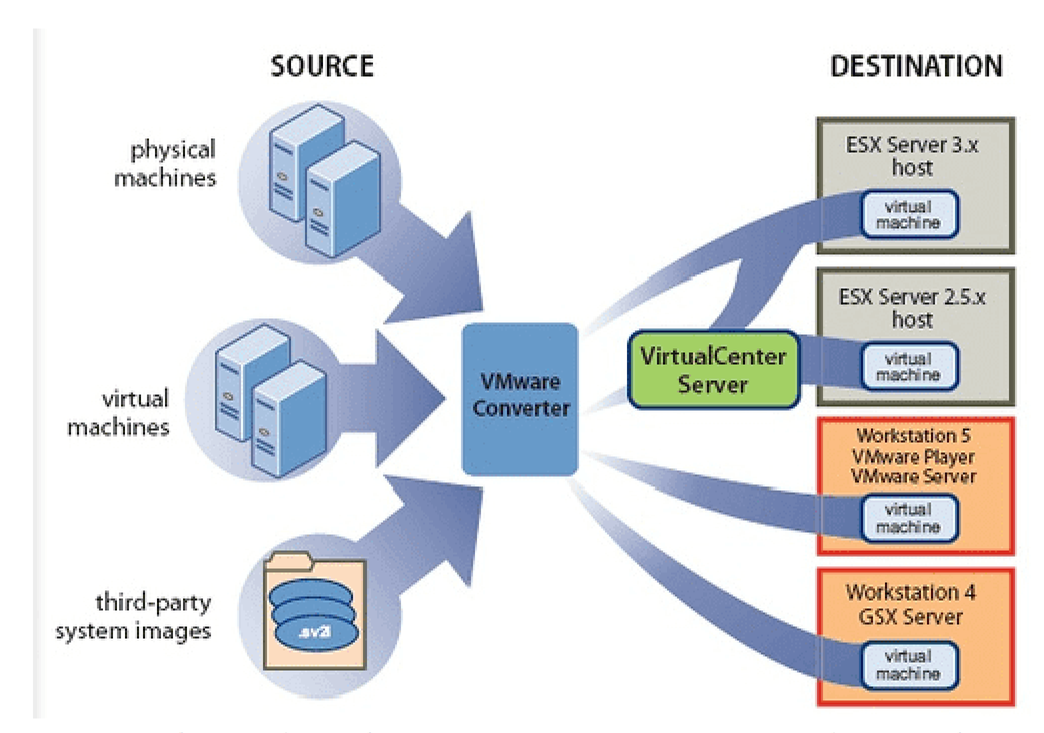
对企业来说，将物理机转换到虚拟机上是一



▲ 以 VMware 官方的测试看，RDM 模式的确有一些速度上的提高

项艰难的工程。这个称为 P2V 的转换（本书稍后的章节会有说明）通常会让原来系统提供的服务中断。对于较大型的系统来说，P2V 的过程甚至会花上好几天才能落实，这让企业将物理机转换成虚拟机的意愿大大降低。

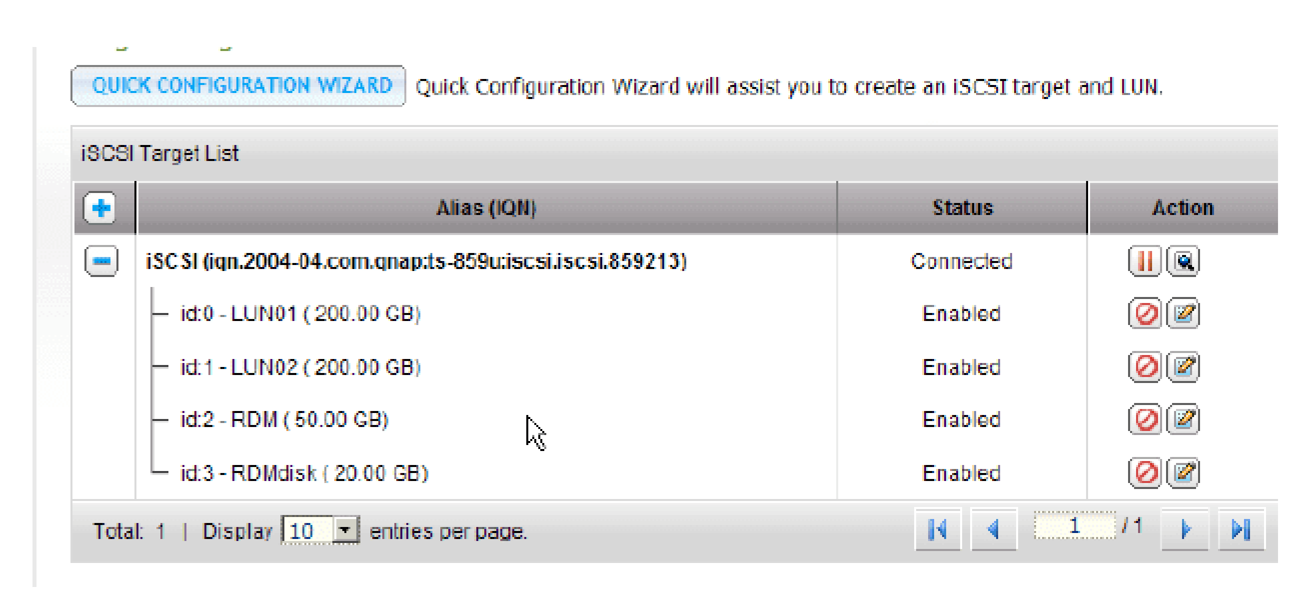
RDM 的最佳使用时机就是在此。如果企业在物理机的操作系统安装时，就使用 HBA 卡或是 PXE Boot 的方式（网卡引导），将操作系统本身安装在远程存储设备的 LUN 之上。当需要进行 P2V 时，再用 RDM 对应到原来操作系统安装的 LUN 上，整个 P2V 的过程就会简化到只有创建 Meta File 的过程，这个时间通常只有几秒钟至一分钟，可以让 P2V 的速度大幅提高。



▲ 物理机转虚拟机的过程，RDM 可以减少停机时间

▶ 使用 RDM P2V 的步骤

1. 企业环境中必须有一个独立的存储设备，并且可以分出逻辑分割区 LUN。



▲ 要有一个物理的 LUN

2. 首先准备一台物理机，这台物理机必须有可连接至 LUN 引导的 FC SAN HBA 卡，或是 iSCSI 卡。能通过网卡引导的 PXE BOOT 也可以。

```
QEMU BIOS - build: 02/08/07
$Revision: 1.174 $ $Date: 2006/10/17 16:48:05 $
Options: apmbios pcibios eltorito rombios32

ata1 master: QEMU CD-ROM ATAPI-4 CD-Rom/DVD-Rom

Booting from Floppy...
Loading ROM image......


gPXE 0.5.1 -- Open Source Boot Firmware -- http://etherboot.org
Features: HTTP HTTPS DNS TFTP iSCSI AoE bzImage Multiboot PXE
net0: 52:54:00:12:34:56 on PCI00:03.0 (open) TX:0 TXE:0 RX:0 RXE:0
DHCP (net0 52:54:00:12:34:56)... ok
net0: 10.254.254.31/255.255.255.0 gw 10.254.254.2
Booting from filename "http://chipmunk.tuntap/images/uniboot/uniboot.php"
http://chipmunk.tuntap/images/uniboot/uniboot.php... ok
/kernels/vmlinuz-2.6.17-14mdv... ok
initrd.img... ok
/kernelmods/2.6.17-14mdv/kernel/drivers/net/mii.ko.gz... ok
/kernelmods/2.6.17-14mdv/kernel/drivers/net/8139too.ko.gz... ok
/kernelmods/2.6.17-14mdv/kernel/net/packet/af_packet.ko.gz... ok
http://rom.etherboot.org/share/mcb30/bootsplash-800x600...
```

▲ 物理机可以从网卡或是 HBA 连接到 LUN 上引导

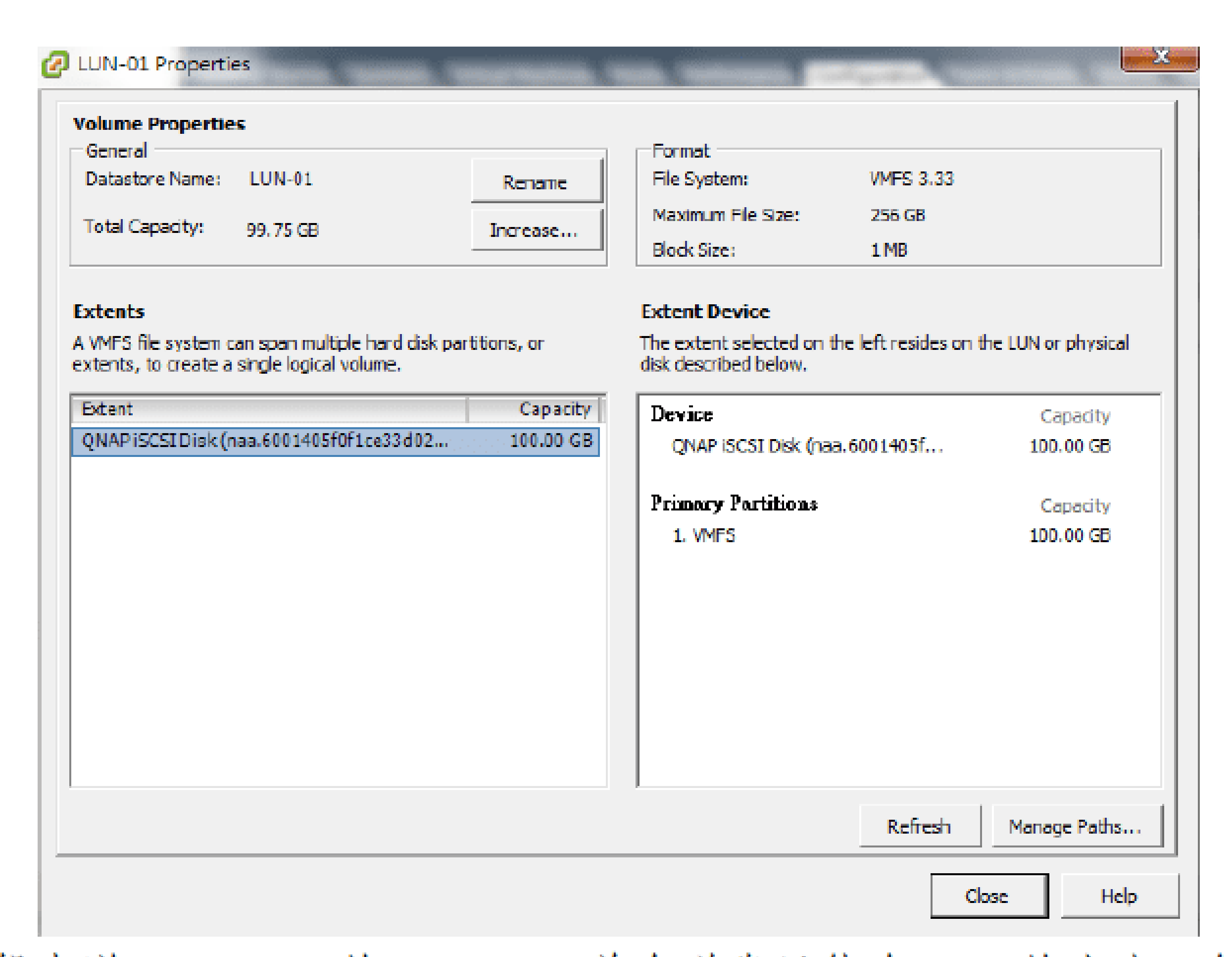
3. 物理机将操作系统通过 HBA 卡安装在 LUN 上，物理机本身并没有硬盘。

当需要进行 P2V 时，使用 RDM 模式，VM 只要创建一个 File Descriptor，将 VM 的硬盘给定到刚才那个 LUN。

此时不需要文件转换，就可以继续可读 LUN 上的 OS，P2V 即落实。此时的耗时将在数十秒以下。

3. 操作 TB 级以上的文件时

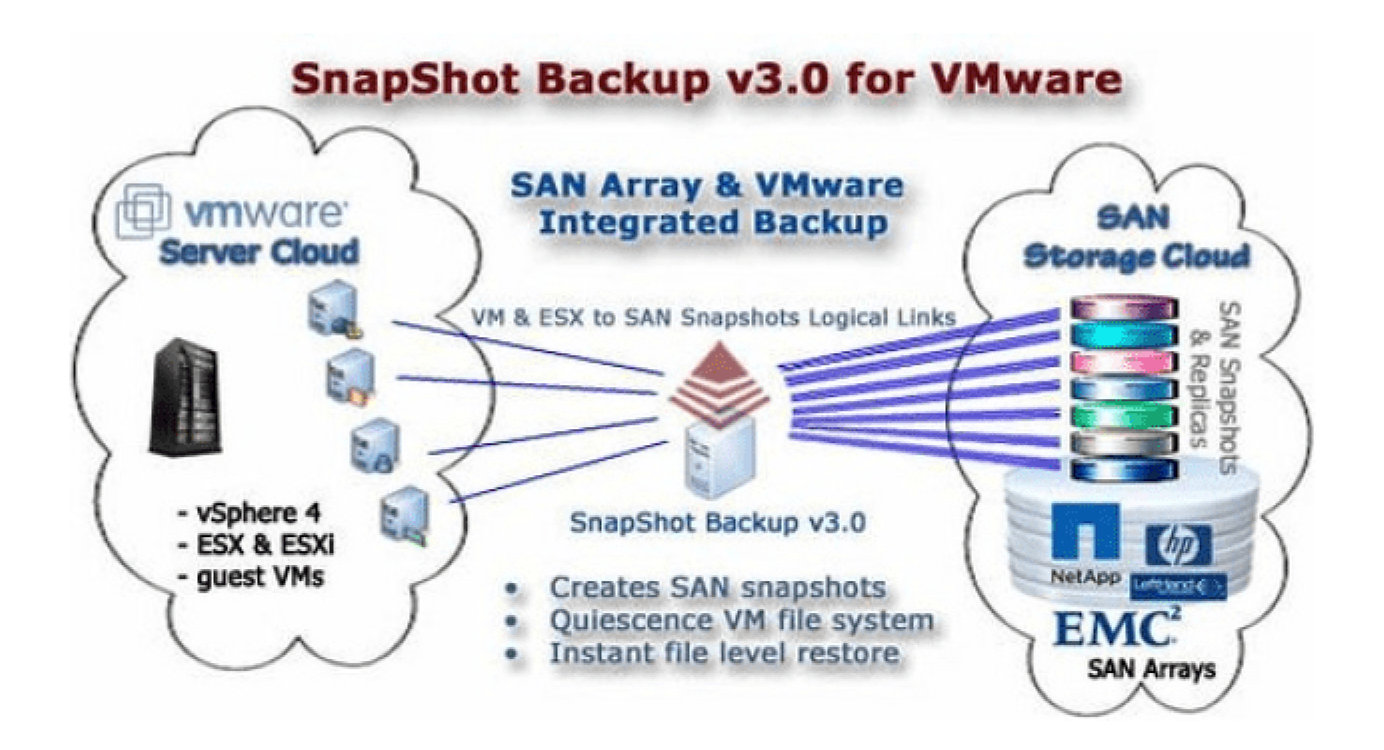
由于 vSphere 的 Datastore 的单个磁盘大小有 2TB 的制约，如果你的 VM 可能会访问 2TB 以上的分割区或是文件时，那么 RDM 将会是唯一的选择。如果你想在 vSphere 中访问超过 2TB 的文件，也可以用分区扩展的方式，读者们可以回到第 12 章引用扩展 vSphere 分区大小的部分。



▲ 一般来说，vSphere 的 Datastore 单个磁盘只能有 2TB 的大小，但可扩展

4. 大量使用 SAN/iSCSI 的工具时

大部分的 SAN/iSCSI 都有相当好用的工具来处理大容量的文件或 LUN，如图像的快照、直接副本、阴影、磁盘镜像等。这些工具能快捷访问 LUN，并且速度可能比 VMware 提供的工具要快捷方便。如果你的使用场合需要常常进行这一类的操作时，RDM 模式将方便这些工具的使用。



▲ 较高级的存储设备都有自身的工具

5. 在 VMware 中使用 MSCS 时

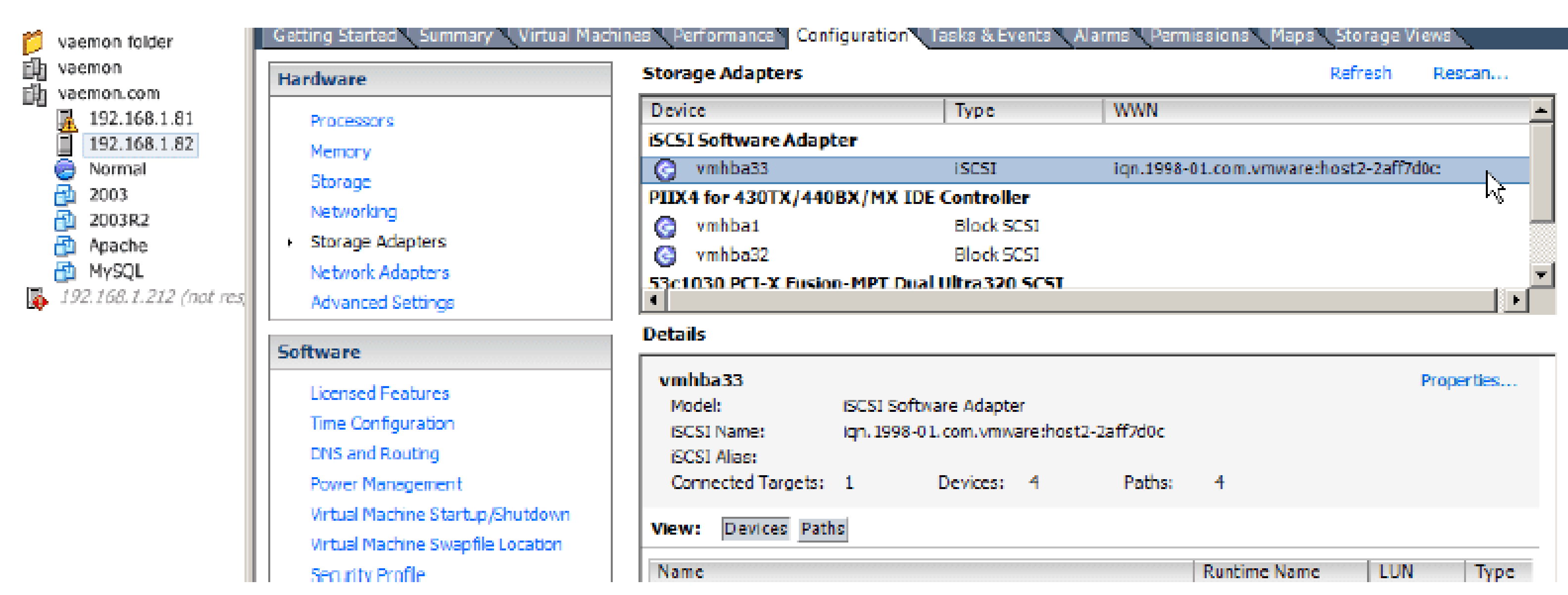
MSCS（微软集群服务，Microsoft Clustering Service）如果在 vSphere 中操作，其安装的机器必须使用 RDM 模式才能进行。

19.1.4 RDM 模式的前提

要在 ESX/ESXi 下使用 RDM 必须满足几个情况，我们就来看看。

1. ESX/ESXi 必须拥有 HBA 卡或激活 iSCSI

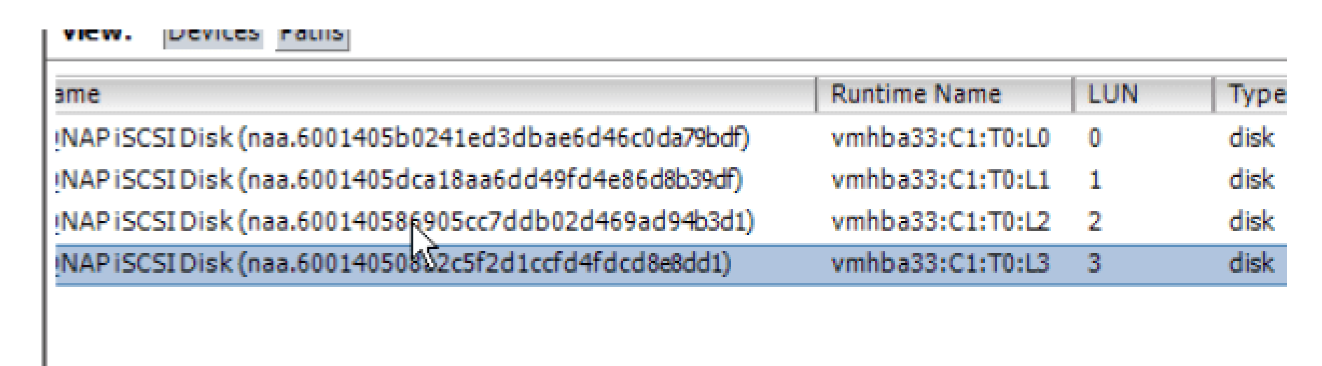
ESX/ESXi 必须要拥有独立的 HBA 卡或是激活 iSCSI。我们可以在 ESX/ESXi 的 Storage Adapter 上看到当前使用的 HBA 卡。如果使用的是传统的网卡，确定 iSCSI 已经被激活了才行。



▲ iSCSI 要被激活了

2. 必须有独立的 LUN

在外部存储设备上，必须有一个独立的 LUN 空间可使用（即在 Storage View 下可看到），但不可以挂成 Datastore。我们可以在 Storage Adapter 卡上看到这片卡可访问的 LUN，在 RDM 模式激活时也可以看到。



▲ 激活的 LUN 不是 Datastore

19.1.5 实践 RDM 模式

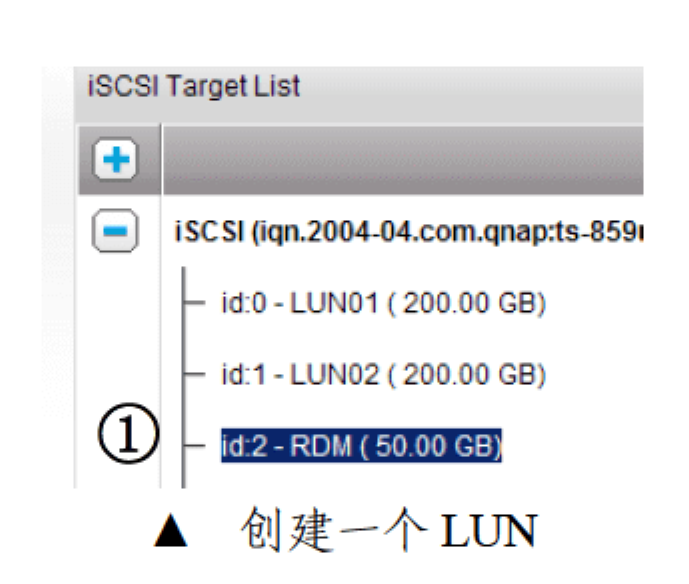
在理解 RDM 之后，接下来就是进入实践了。实践 RDM 再简单不过，一般就是在创建 VM 时选择硬盘的部分，当然你也可以替任何一个 VM 添加 RDM 模式的磁盘，我们就来看看步骤。

1. 创建 RDM 的 VM

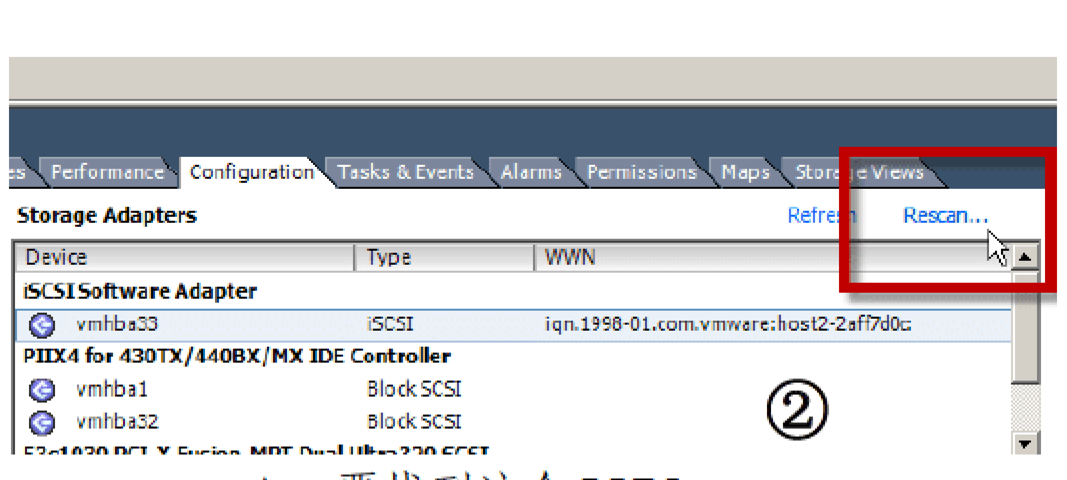
我们还是创建一个新的 LUN，并且让这个 LUN 能被 ESX/ESXi 主机辨别，之后在 VM 的创建过程中新建即可。

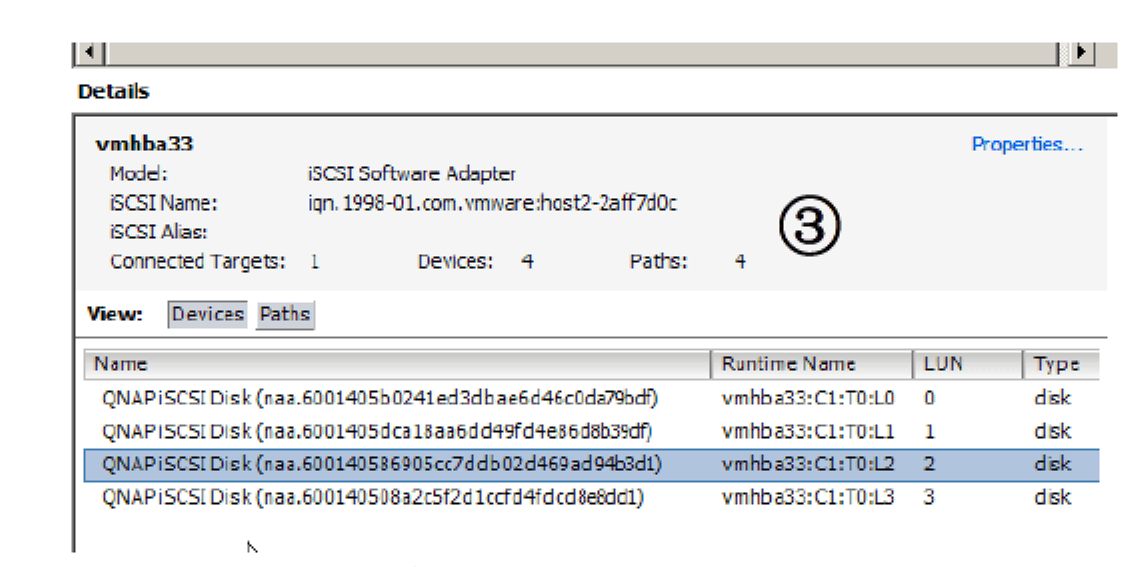
▶ 创建以 RDM 为主的 VM

1. 首先我们先进入管理iSCSI的界面，并且创建一个LUN。
2. 接着进入 vSphere Client，在主机的 Storage Adapters 的地方单击 Rescan 链接。
3. 看到这个 iSCSI 的所有 LUN，其中最后一个 LUN 是能被找到。



▲ 创建一个 LUN

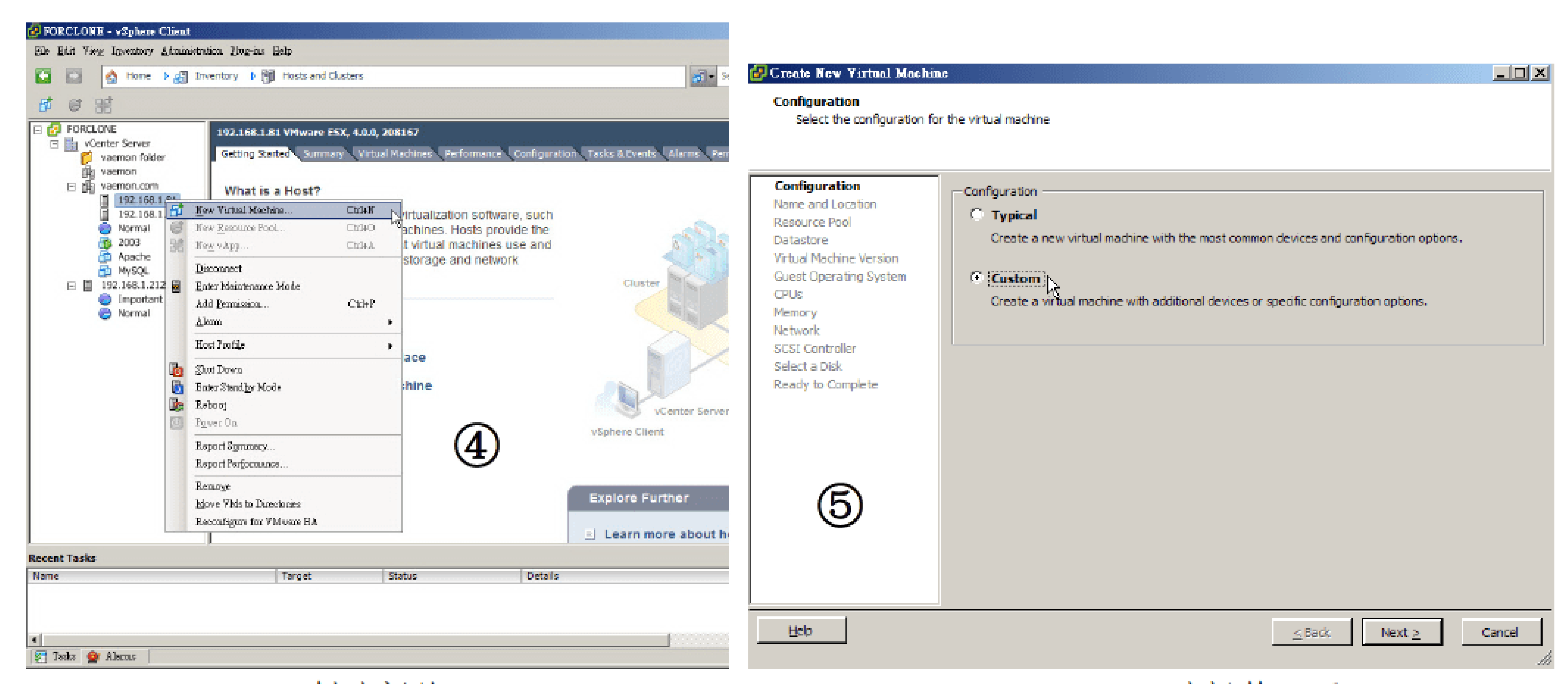




▲ 要找到这个 LUN ▲ 找到了这个被找到的 LUN 不可以加入 Datastore

4．接着进入创建 VM 的画面。创建新的 VM。

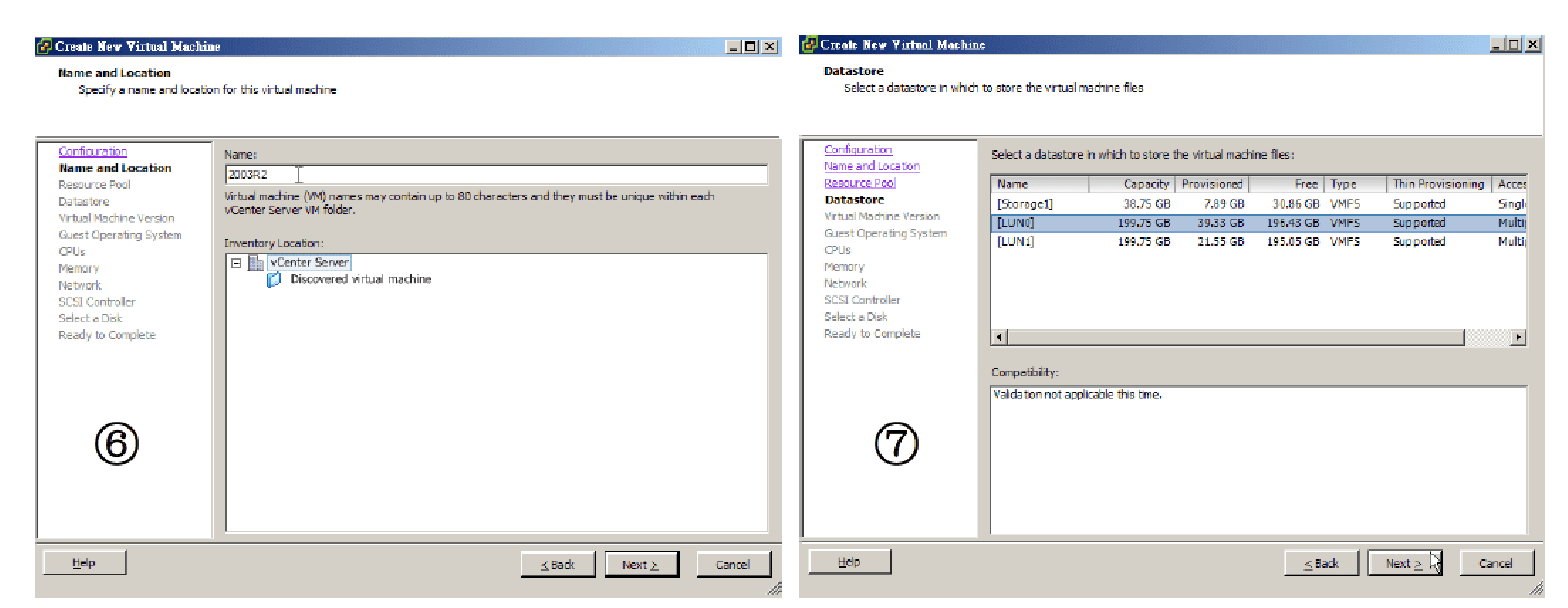
5．一定要单击 Custom 单选按钮，要不然会无法进入 RDM 的选项。



▲ 创建新的 VM ▲ 选择第二项

6．接下来是选择 VM 的名字。

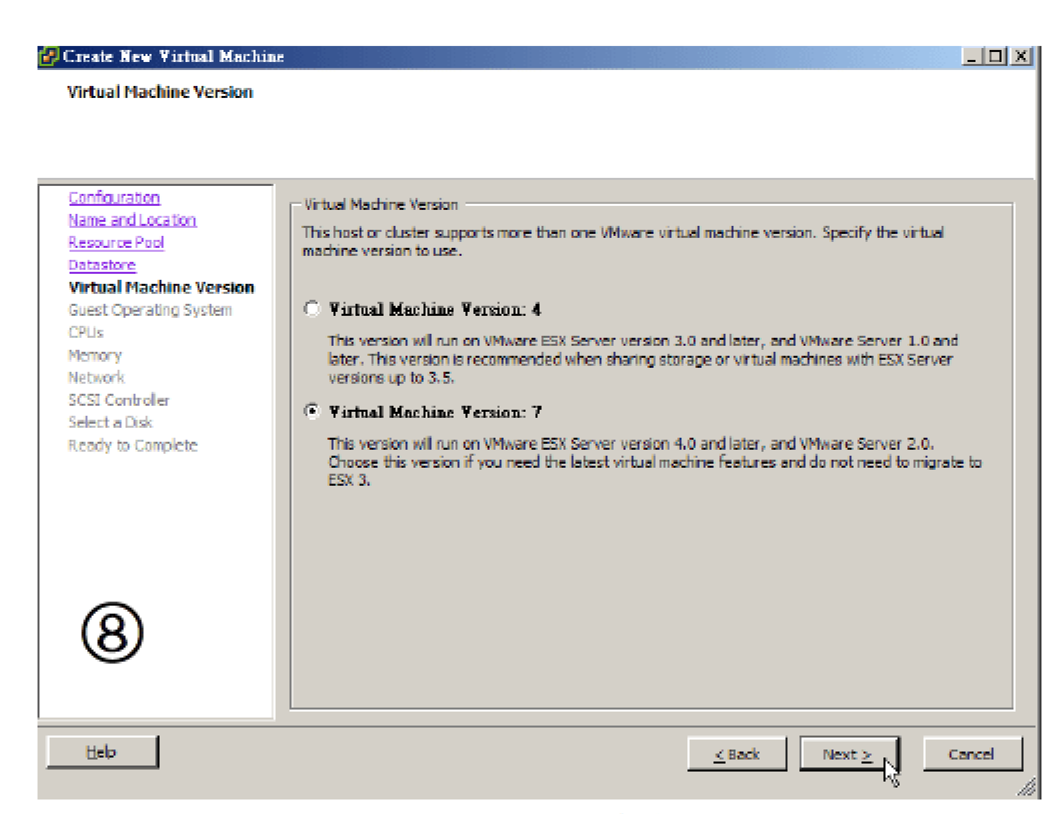
7．然后是选择置放 VM 的地方，我们会选择独立存储设备，以方便 VMotion 或是其他的 vCenter 功能。



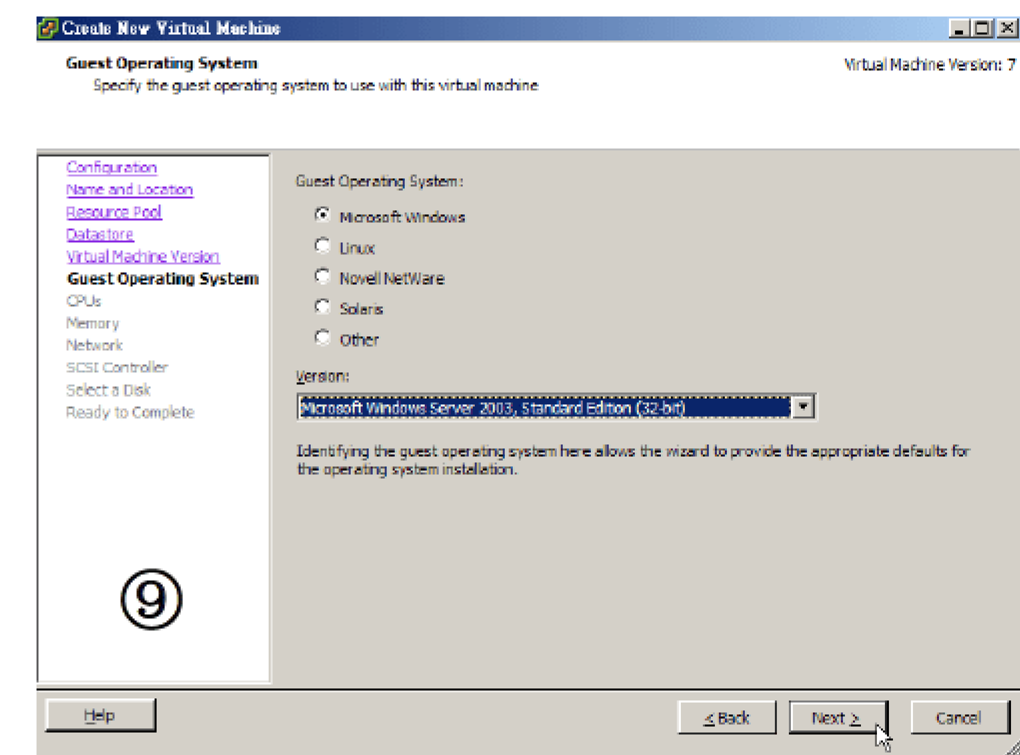
▲ 这里是创建 VM 的方式，就复习一下 ▲ 将 VM 放在独立存储上

8．接下来选择 VM 的兼容性，这里选择版本 7。

9．选择操作系统。



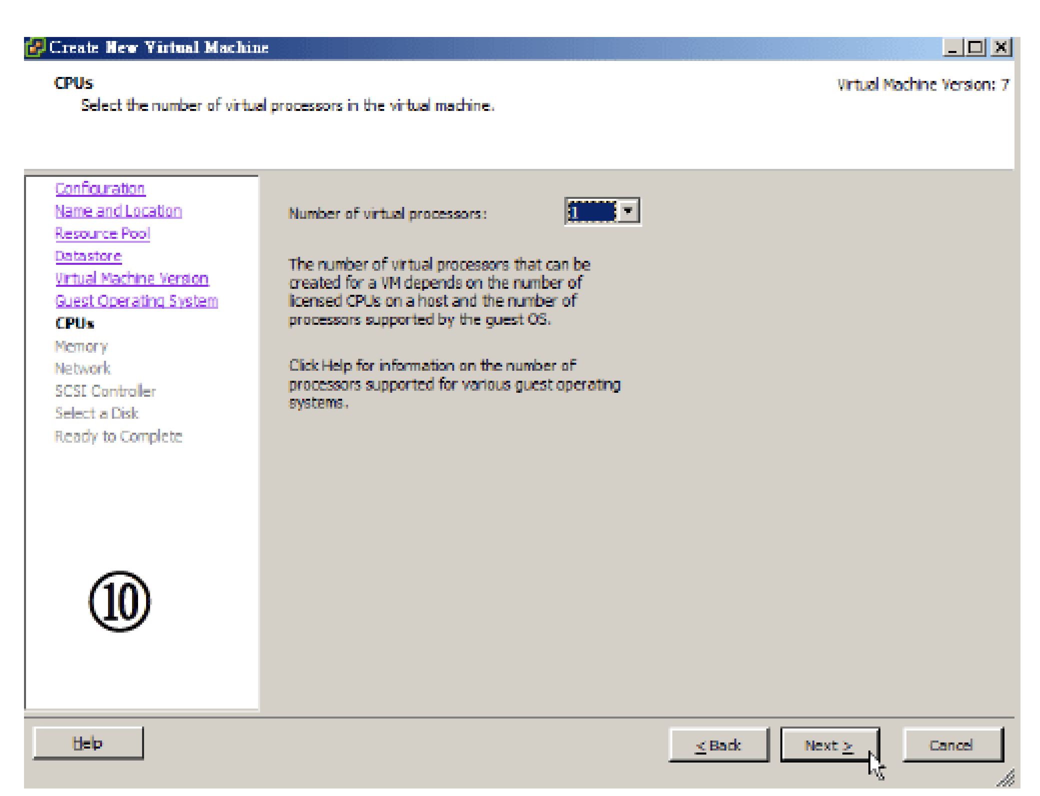
▲ 选择版本兼容性



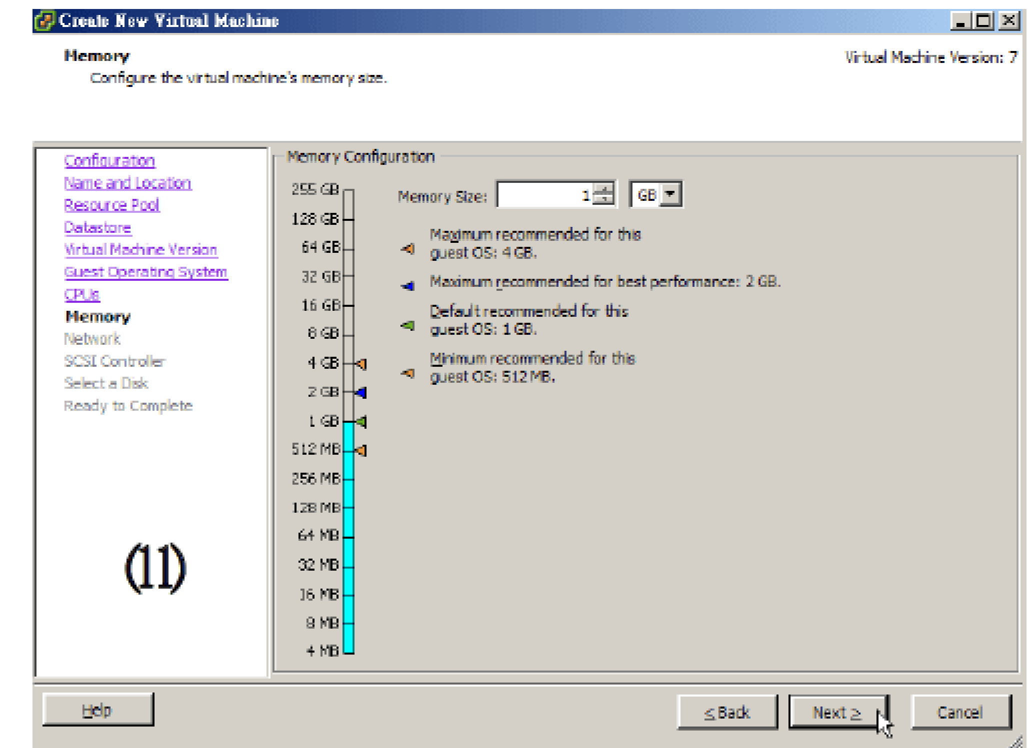
▲ 选择操作系统

10．选择处理器数量。

11．选择内存数量。



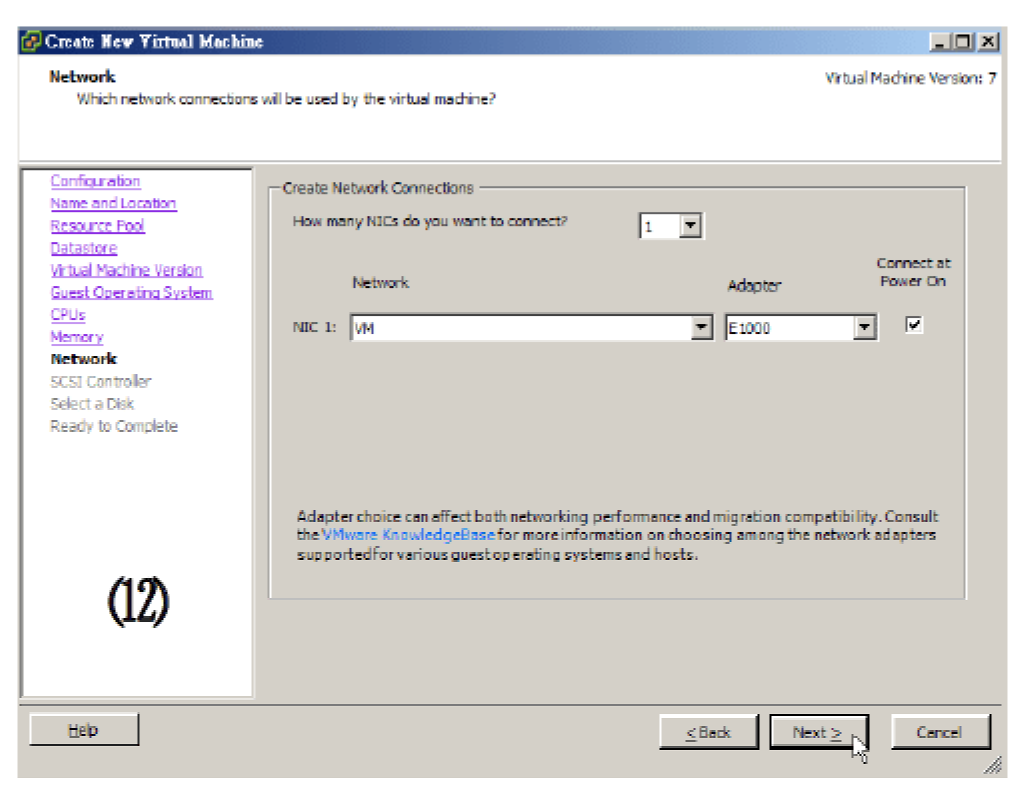
▲ 选择 CPU 数量



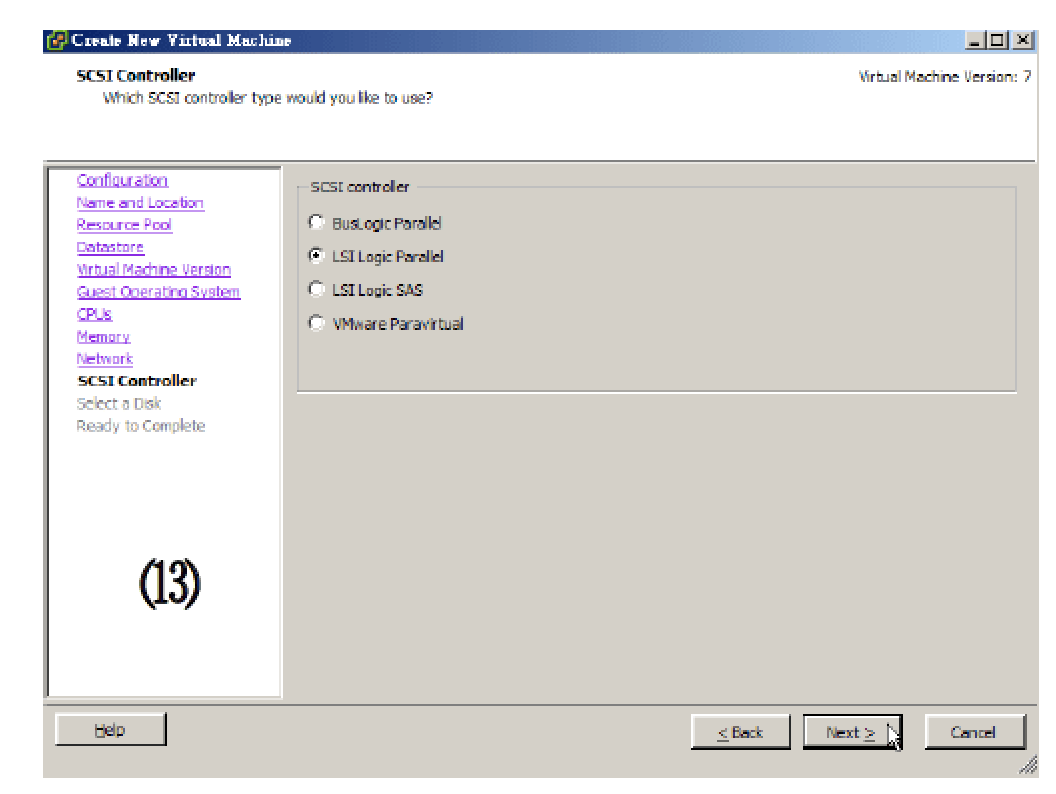
▲ 选择内存

12．选择网卡用哪一个 Port Group。

13．接下来要选择 SCSI 厂牌。要注意的是，RDM 模式也必须给定一个 SCSI 卡给 VM 作为对应。



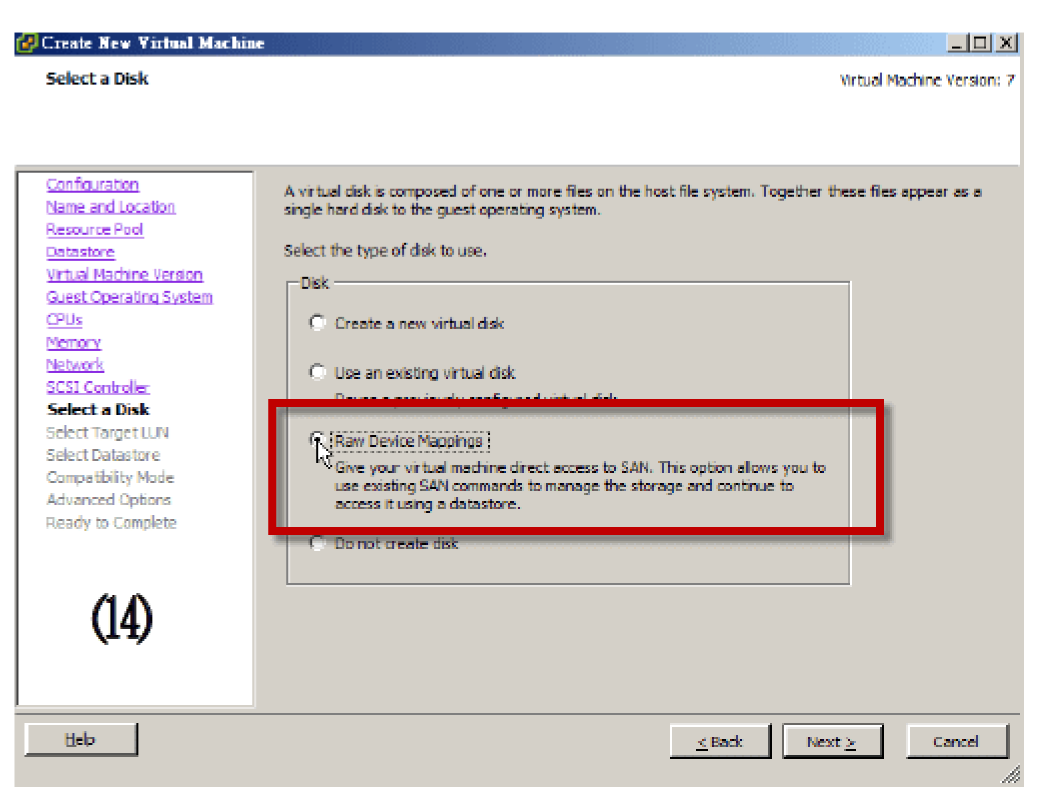
▲ 选择网卡



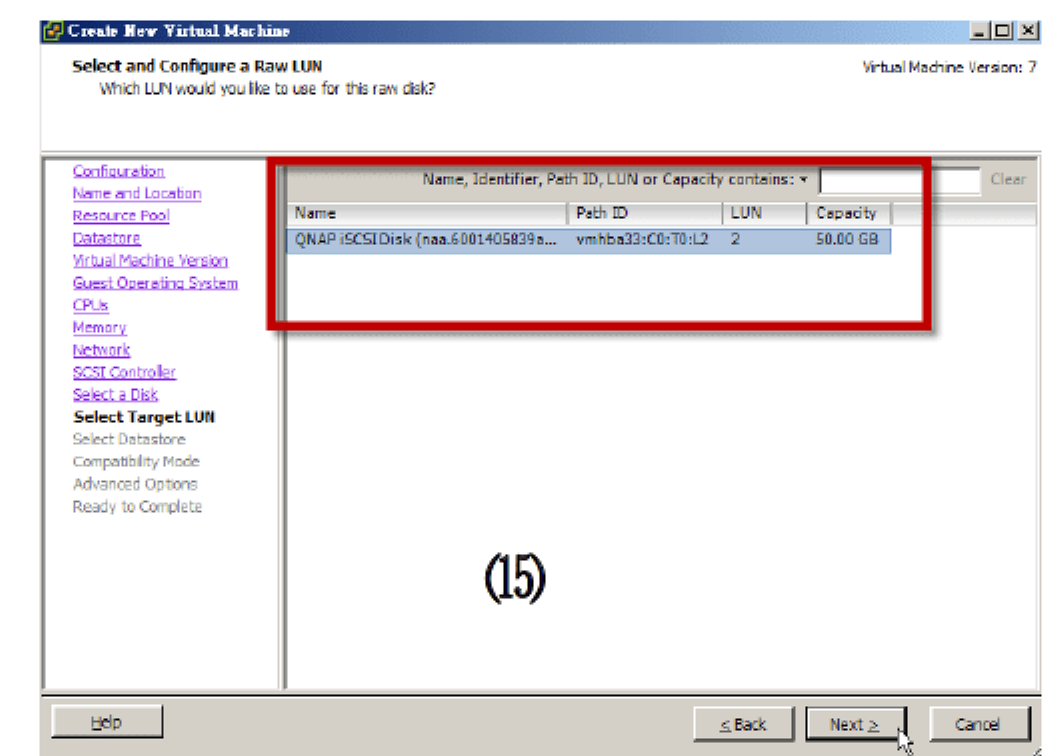
▲ RDM 模式也必须用虚拟的 SCSI 卡

14．接下来是选择硬盘的连接方式，当然就是 RDM。

15．选择刚才挂上的 LUN。



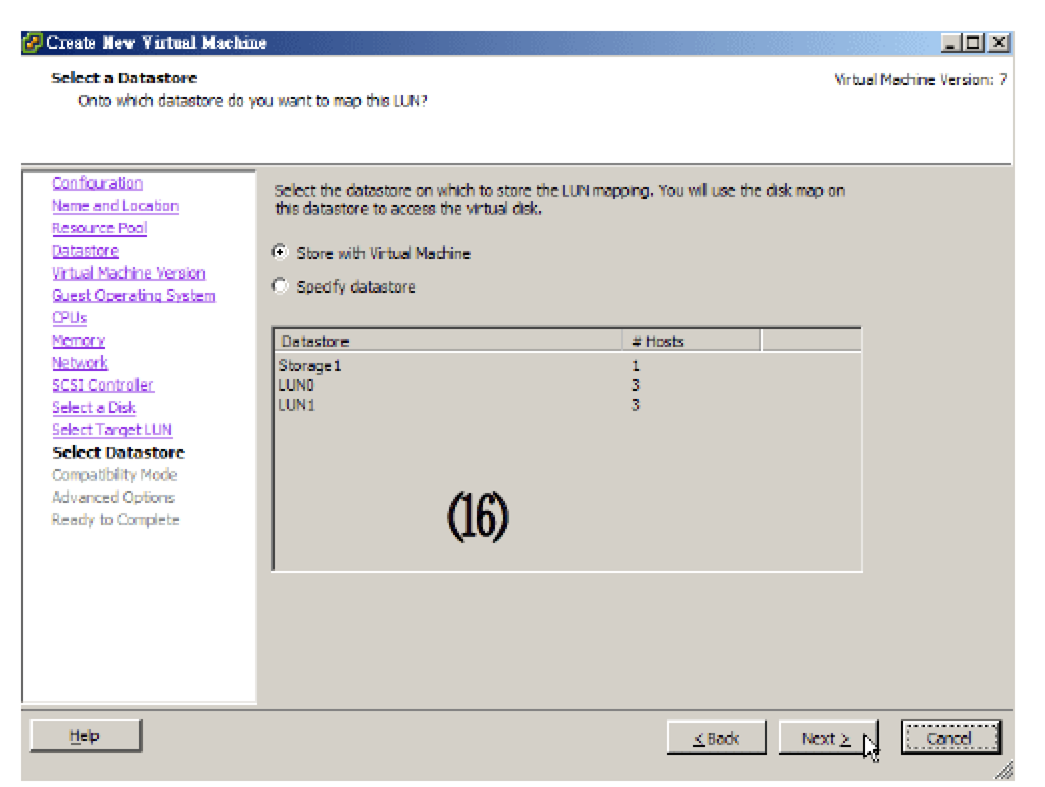
▲ 这里是热点，要选择 RDM 模式



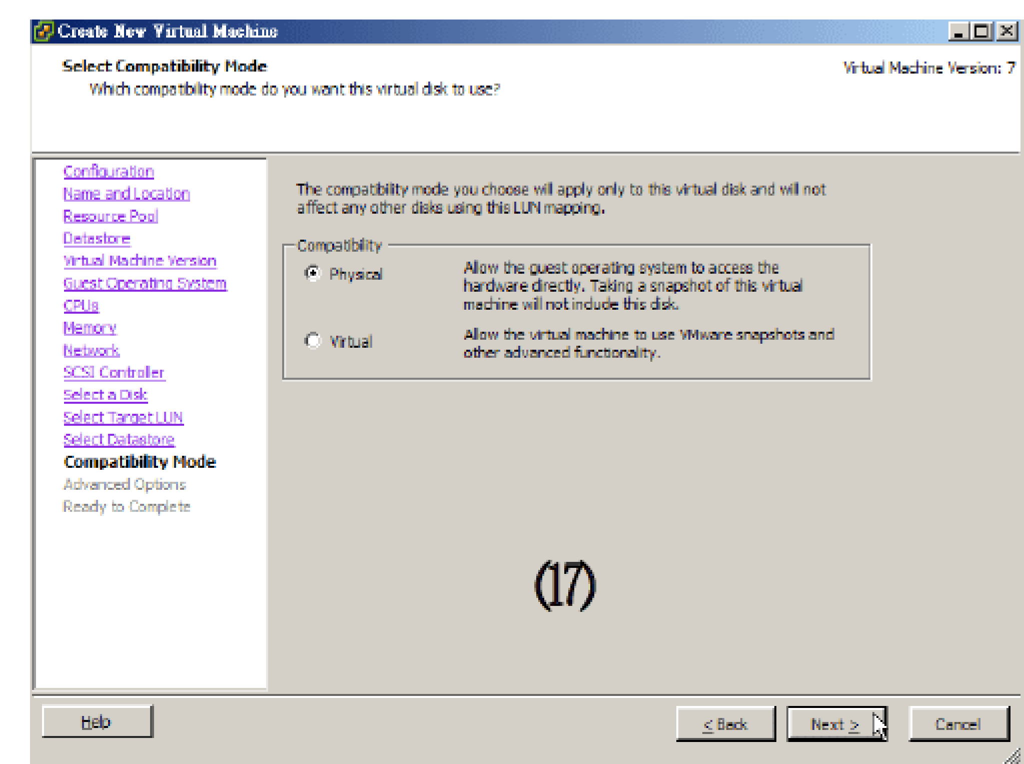
▲ 选择刚才挂上的 LUN

16．接下来是选择 VMDK 置放的位置。在 RDM 模式中，VMDK 已成为一个文本档，作为对应用。

17．接下来是选择 RDM 的兼容模式，前面提过有物理兼容和虚拟兼容，如果你还需要快照等功能，就选择第二项。



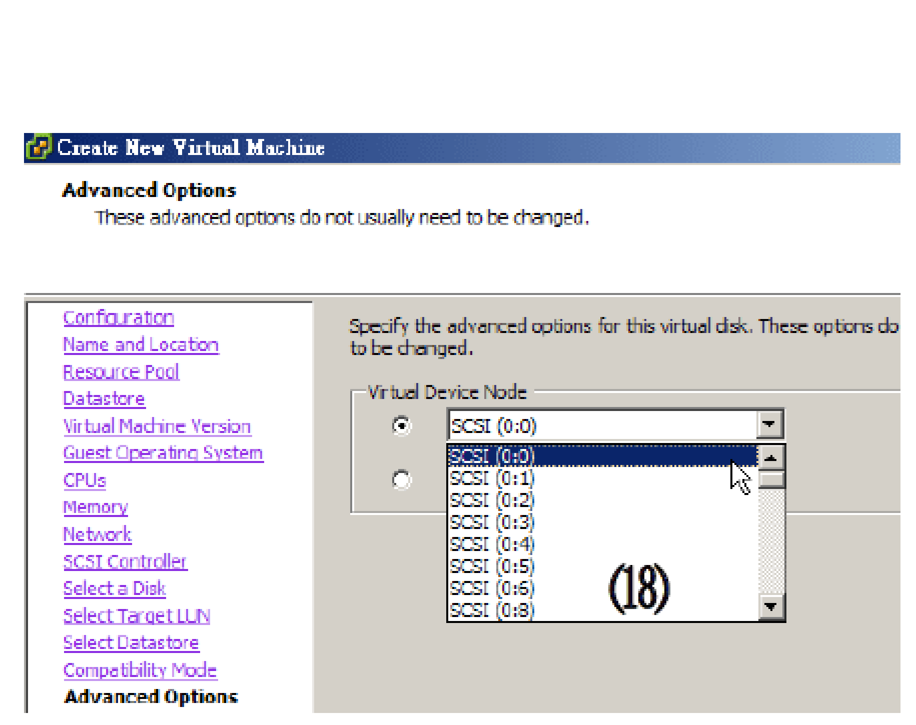
▲ 还是有 VMDK，但已经是文本档了



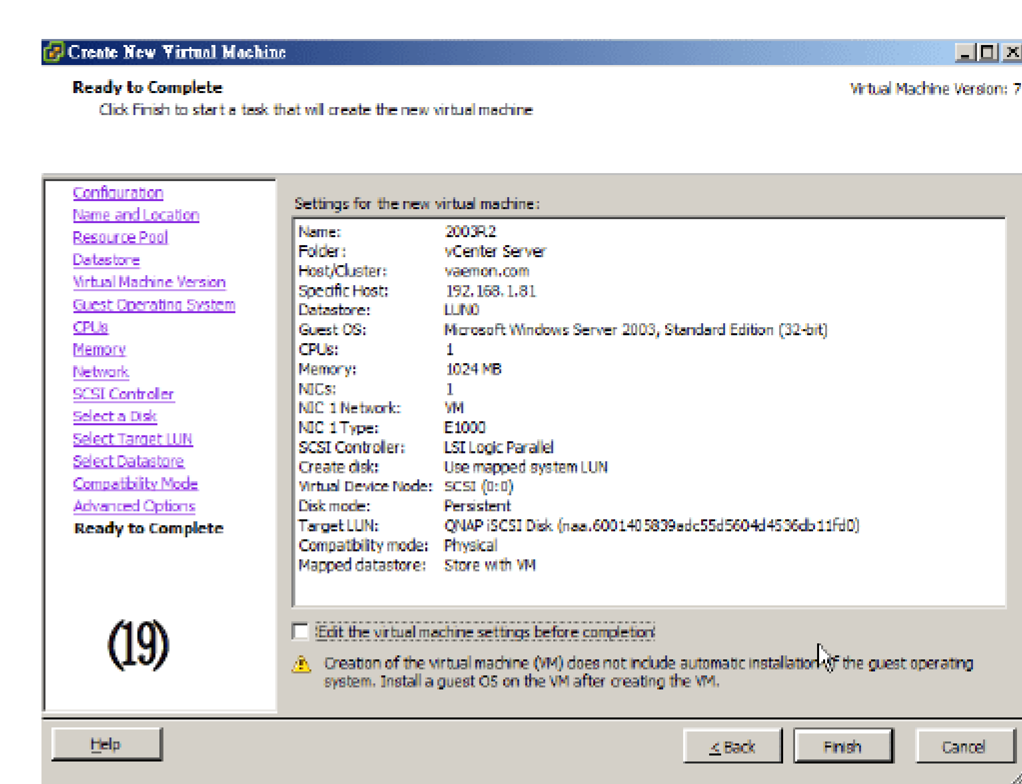
▲ 选择 RDM 的兼容模式

18．接下来还是选择虚拟 SCSI 的端口，就算是 RDM，也需要一个对应。

19．当选择完之后，可以在总结画面中看到 LUN 和 RDM 的给定。

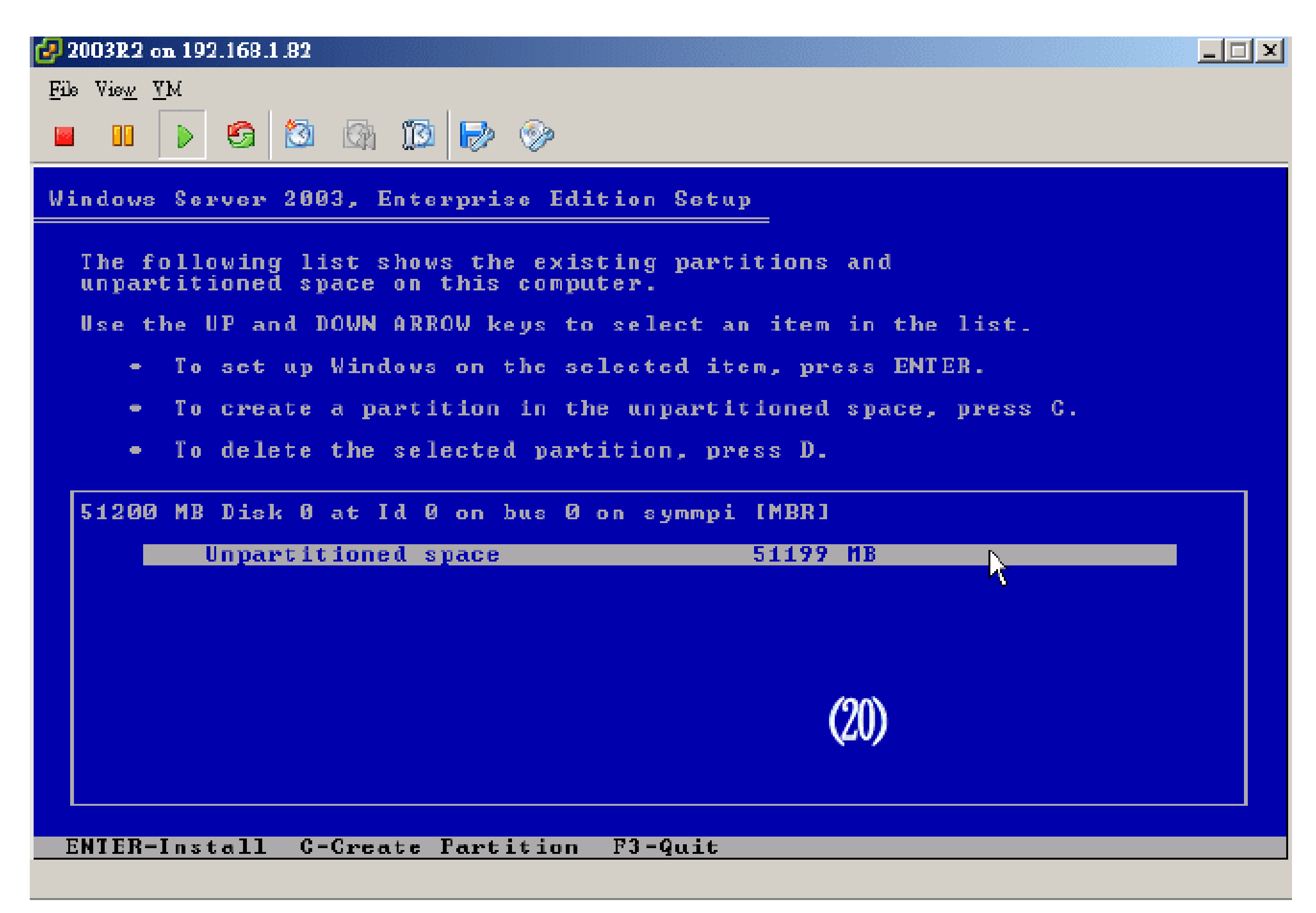


▲ 选择 SCSI 的端口



▲ 看到是 RDM 模式了

20．引导后，果然可以看到这个硬盘。



▲ 这是在安装画面时看到的硬盘，就是刚才创建的 RDM LUN

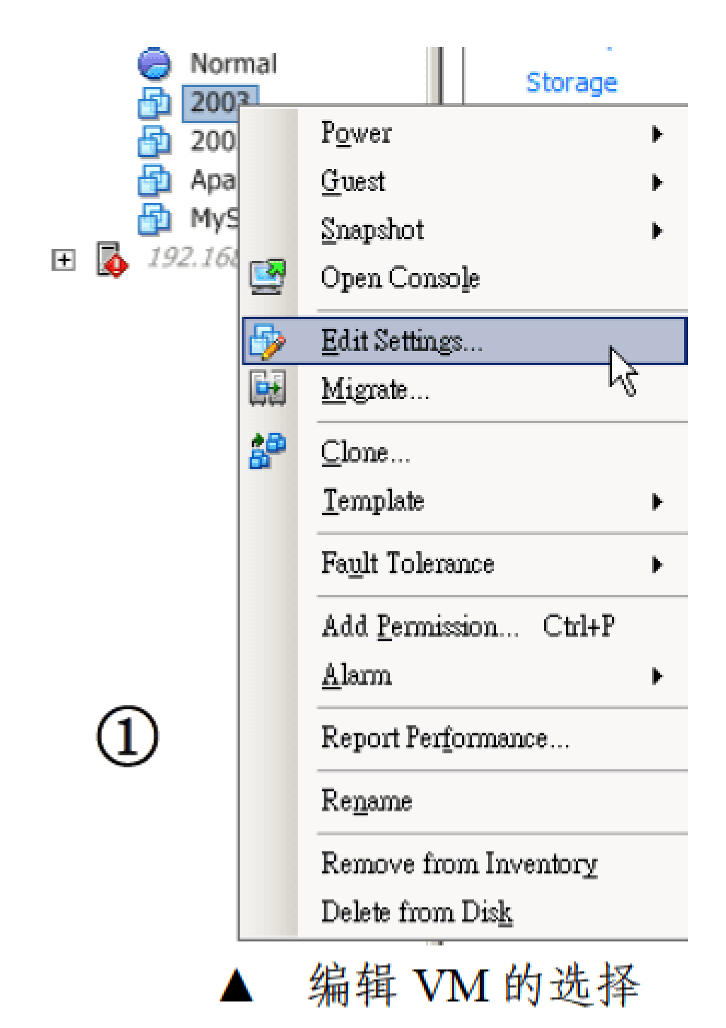
2. 替已创建的 VM 加入 RDM 硬盘

我们也可以替现在的 VM 添加一个 RDM 的磁盘。只要在 Storage Adapter 中找到这个 LUN 即可。下面就是详细步骤。

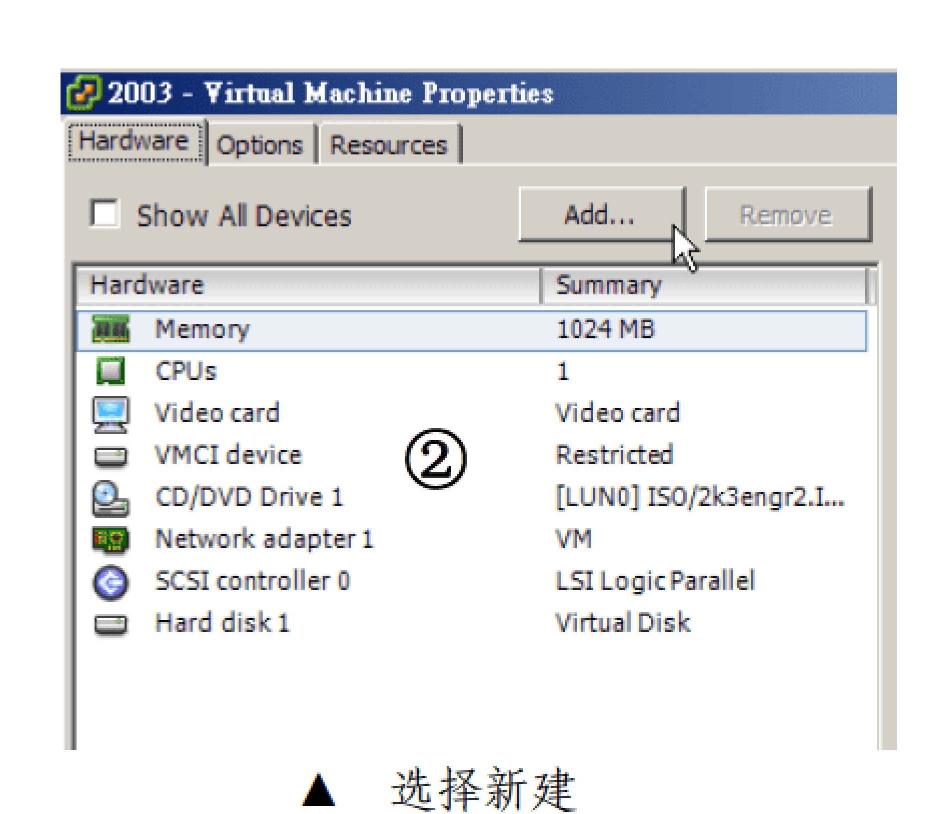
▶ 替现有的 VM 添加 RDM 硬盘

1．首先确定 LUN 已经可被 ESX/ESXi 找到。

2．在要添加 RDM 的 VM 右击，选择 Edit Settings 选项。单击 Add 按钮。



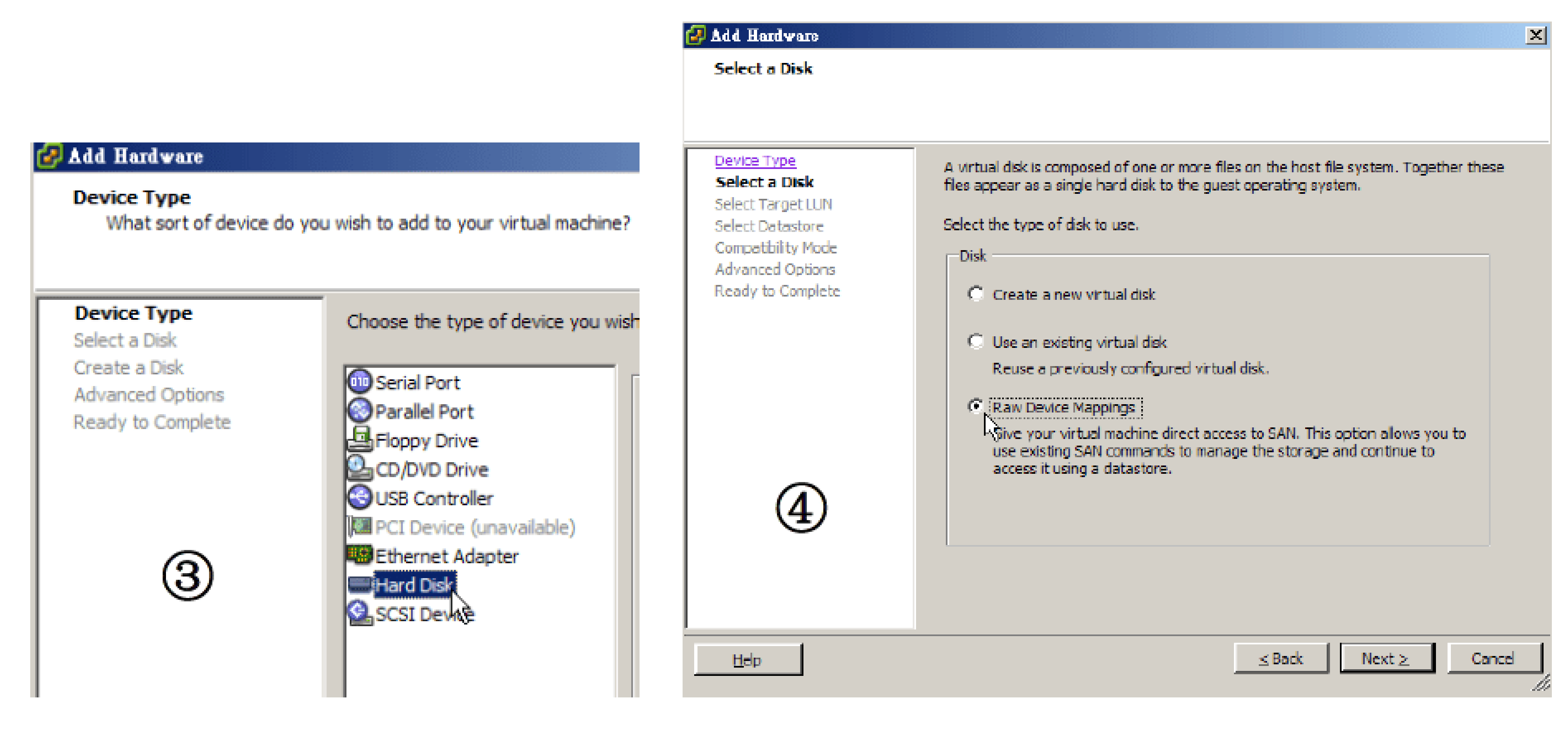
▲ 编辑 VM 的选择



▲ 选择新建

3．选择 Hard Disk 选项。

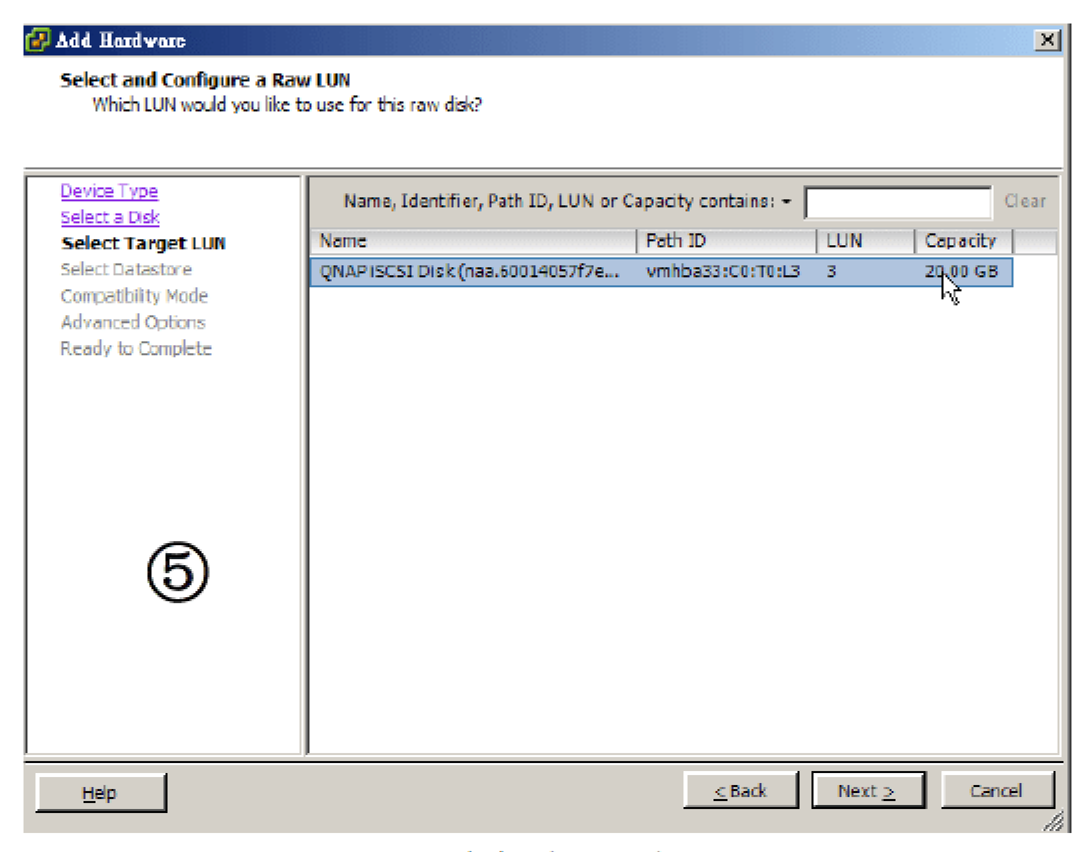
4．单击 Raw Device Mappings 单选按钮。



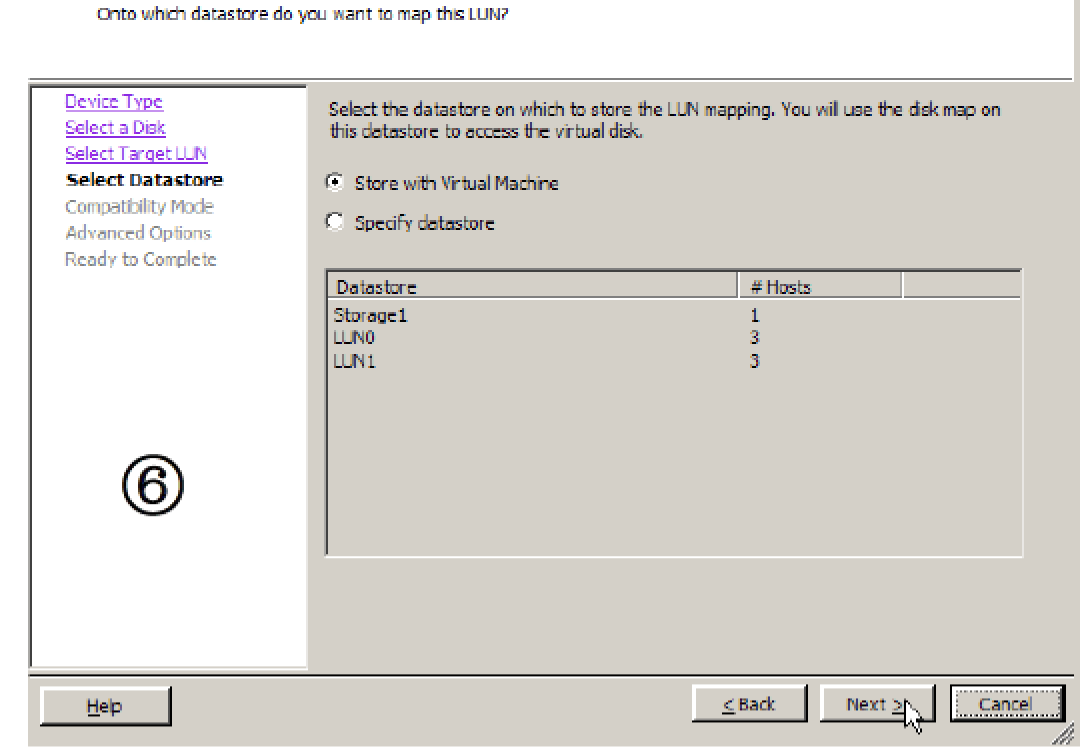
▲ 选择硬盘　　　　▲ 这里也是选择 RDM

5. 选择要加入的 LUN。

6. 选择 VMDK 的置放处。



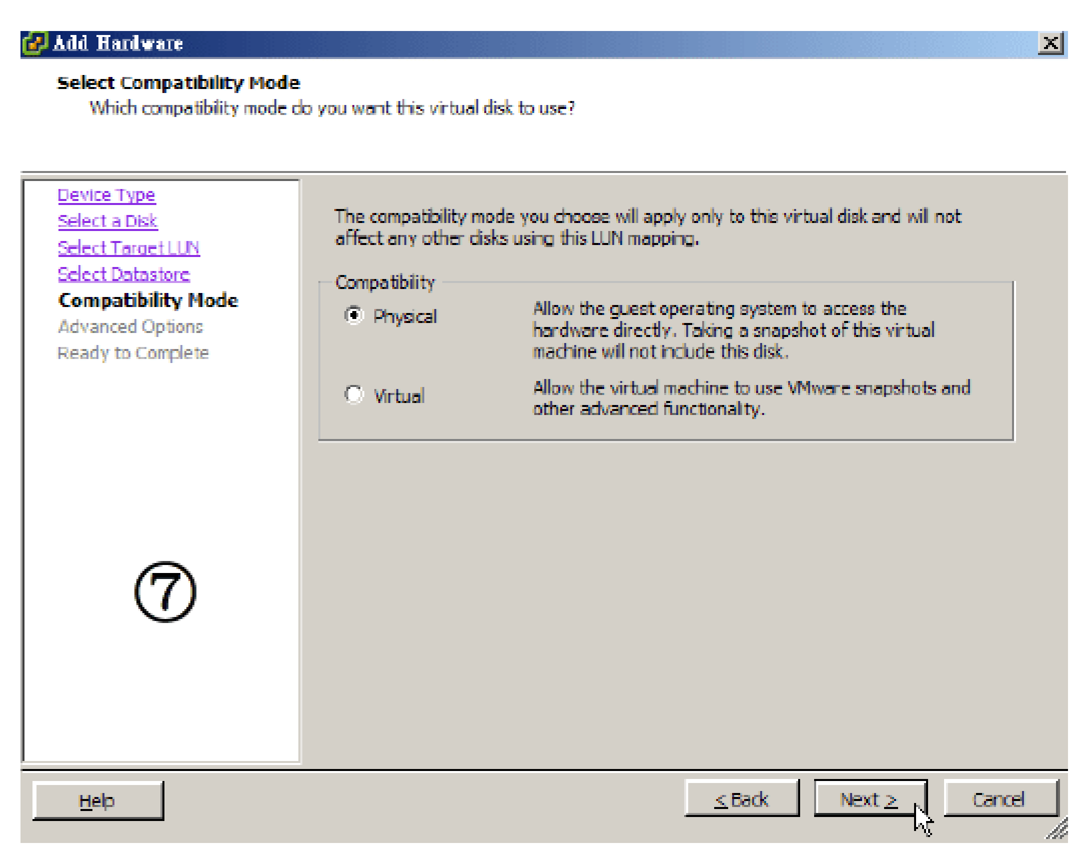
▲ 选择加入的 LUN



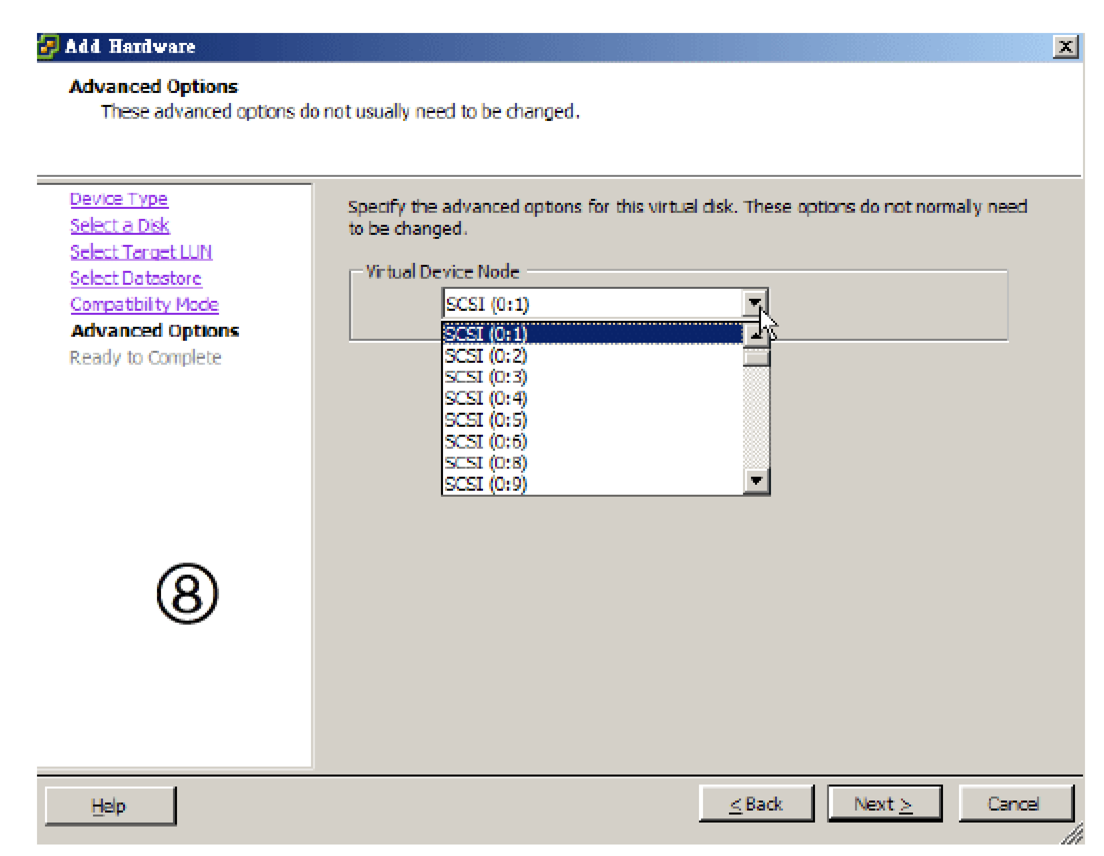
▲ 选择 VMDK 的置放处

7. 选择 RDM 的兼容模式。

8. 选择 SCSI 端口。

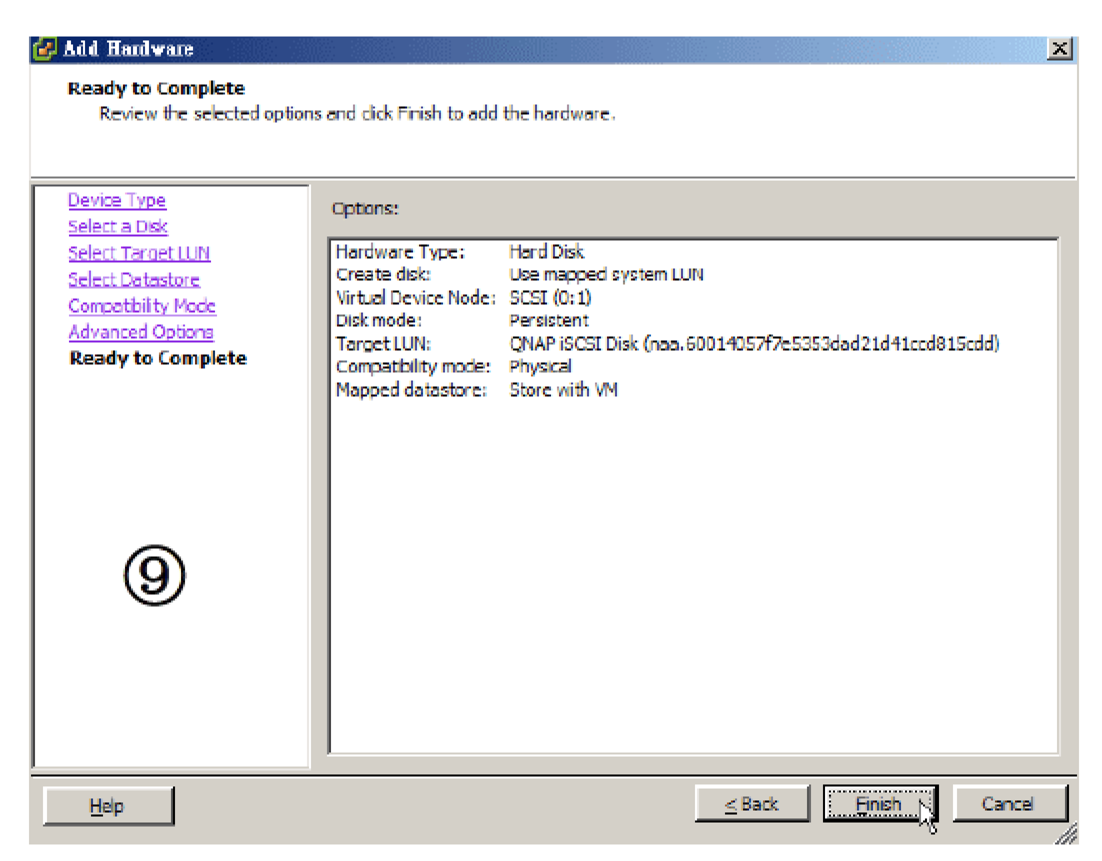


▲ 一样是兼容模式的选择



▲ 选择 SCSI 端口

9. 弹出总结画面，单击 Finish 按钮即落实。



▲ 新建落实

> **注意**
>
> **没有快照的 RDM**
>
> 当你选择了物理兼容模式的 RDM 之后，你的 VM 就失去了快照能力，当然你还是可以使用存储设备的快照功能，但并没有 vSphere 中的快照方便，因此在创建 RDM 时，必须要确定快照功能能被取代，或是根本不需要快照功能。

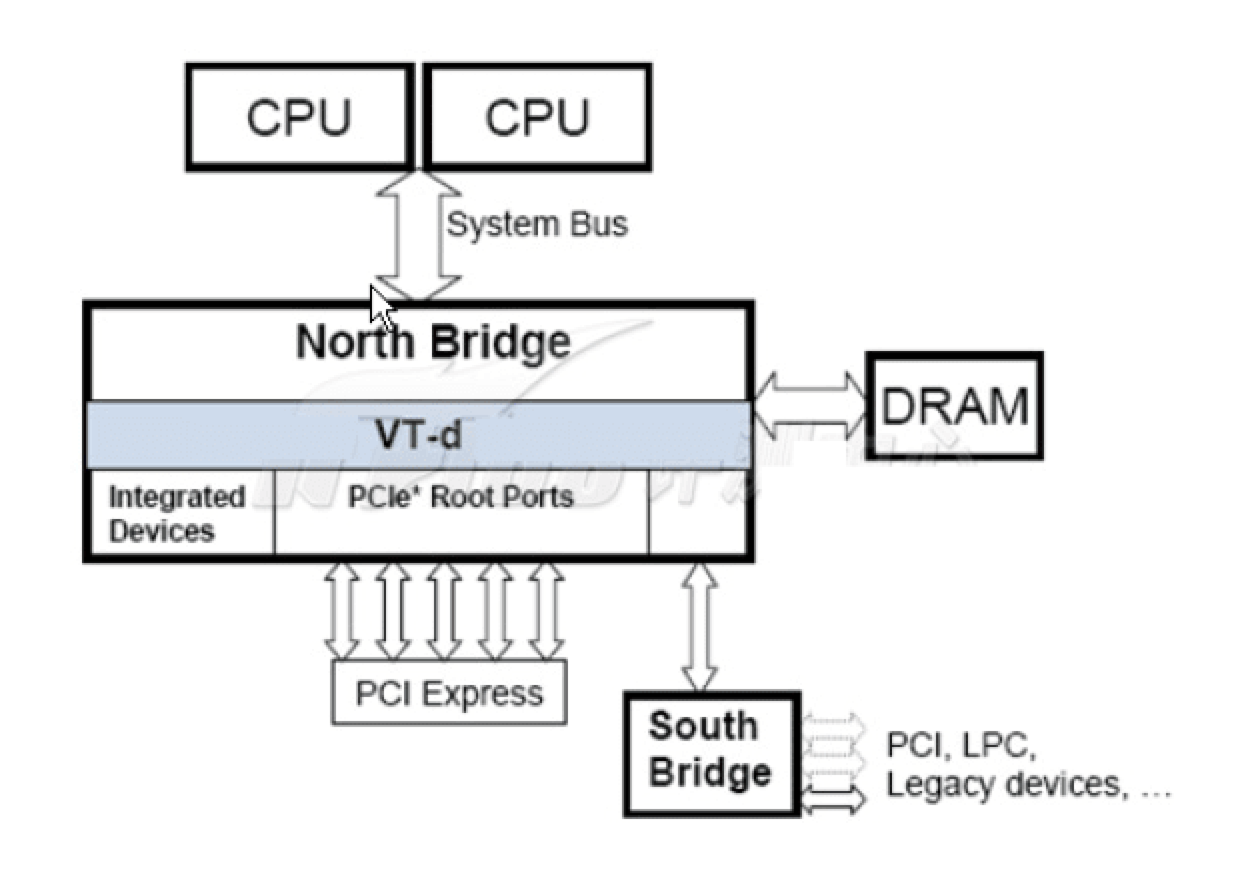
19.2 理解 vSphere 的直接硬件访问

将物理机转换到虚拟机有说不完的好处，但就是有无法转换到 VM 的物理机，主要原因是硬件的兼容性。举例来说，有些服务一定要使用特殊的硬件，但转换到 VM 上之后，所有的硬件都是“虚拟”的，因此这个服务就无法使用了。

在虚拟化的产业中，这个问题的解决方式不是只靠 VMware 这一类的 Hypervisor 厂家能独立落实的。硬件及操作系统厂家也必须提出解决专案。因此当 vSphere 推出之际，主流的硬件厂家（包括 CPU 及芯片组）也提供了对应的专案，这样做的目的，就是 vSphere 中的 VMDirectPath I/O pass-through 功能。

19.2.1 直接硬件访问模式是什么

直接硬件访问模式是一项新的 VM 技术，主要的目的就是让 VM 上的操作系统不需要通过 Hypervisor，直接可以访问硬件，也就是 I/O 设备。但这个直接的访问，也是需要通过一个仲裁的机制，这就需要靠硬件的支持才能落实的。当前提供直接硬件访问功能的硬件包括 Intel 的 VT-d 及 AMD 的 IOMMU。其主要的目的用处就是不再使用 Hypervisor 来类比 I/O 设备的访问，而是让 VM 真正能使用到物理的硬件设备。



▲ 这是标准的直接硬件访问示意图

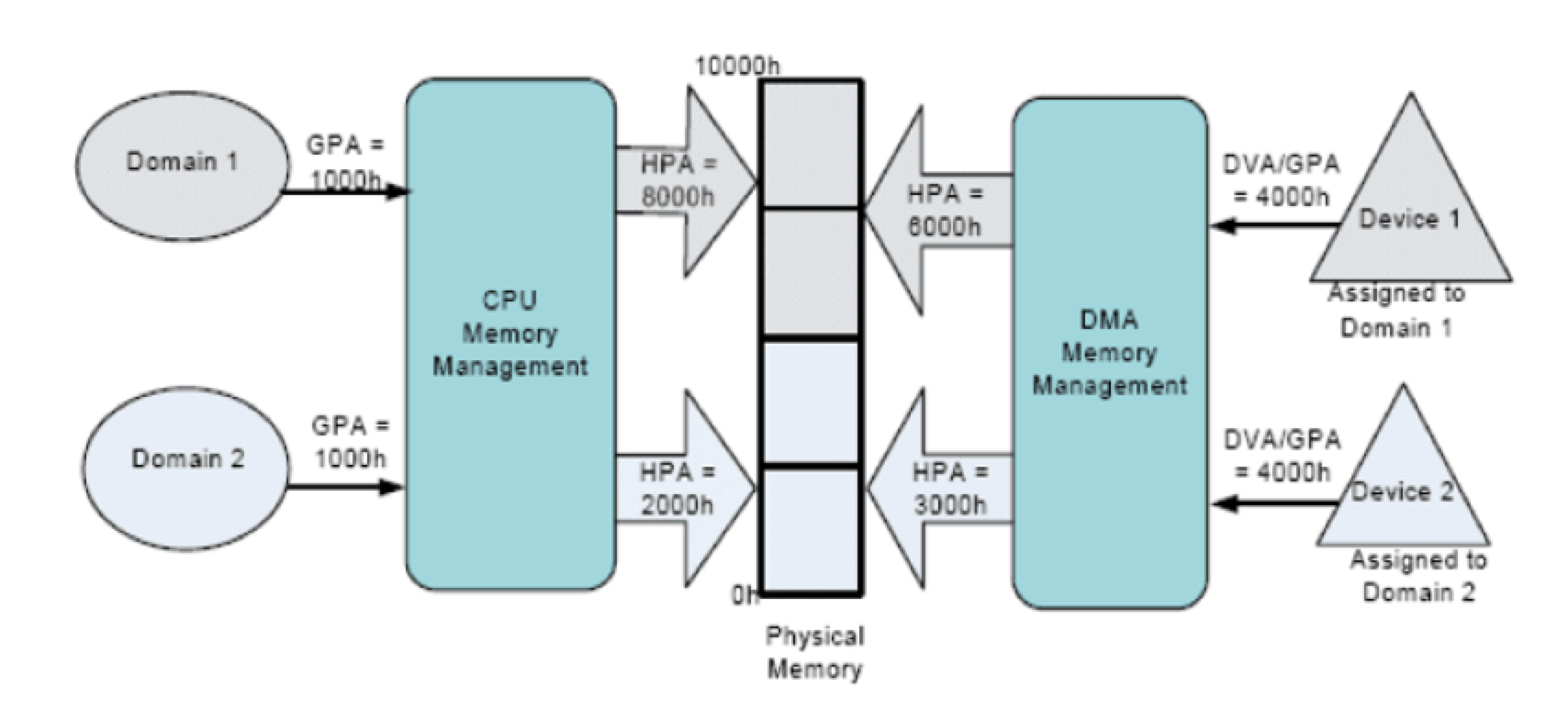
1. 理解直接硬件访问模式的原理

类比 I/O 设备是 Hypervisor 的一个重要功能。Hypervisor 对客户端模拟一个 I/O 设备，通过完全类比设备的功能，VM 可以对应到真实驱动程序，这个方式可以提供透明的兼容性，但是显然这种类比会影响到性能。vSphere 在提供显示适配器、网络设备等功能时，就是使用这个方式，这个方式顾及了整个 VM 的兼容性，但在效能上几乎完全不讲究。

由于现在的 I/O 设备虚拟化主要是采用类比方式或者软件通信端口方式，因此性能上很容易成为“瓶颈”，毕竟传统的机器上，I/O 设备都很容易成为“瓶颈”，因此 Intel 就适时提出了 Intel Virtualization Technology for Directed I/O，简称为 Intel VT-d。而另一个 CPU 厂家 AMD 也提出了相同的概念，称为 IOMMU。

2. 直接硬件访问的热点

在一般没有虚拟化的计算机中，直接访问硬件就是直接访问该硬件的内存位置，称为直接内存访问（Direct Memory Access，DMA）。不管 Intel 的 VT-d 或是 AMD 的 IOMMU（I/O memory management units）都使用了集中方式管理所有的 DMA。除了传统的内部常用硬件的 DMA，还包括如 AGP GART、TPT、RDMA over TCP/IP 等这些特殊的 DMA，通过在内存地址范围来区隔设备并不容易实现 DMA 隔离，因此 VT-d 通过补丁设计的 IOMMU 架构，实现了多个 DMA 保护区的存在以实现 DMA 虚拟化，此技术称为 DMA Remapping。



▲ 也称为 DMA Remapping

另外 Intel VT-d 技术和 AMD 的 IOMMU 是一种基于北桥芯片或是 Memory Control Hub 的硬件辅助虚拟化技术，通过在北桥中内置提供 DMA 虚拟化和 IRQ 虚拟化硬件，实现了新型的 I/O 虚拟化方式，Intel VT-d 能够在虚拟环境中大大地提升 I/O 的靠谱性、灵活性与性能。

VT-d 的关键是解决了 I/O 设备与虚拟机数据交换的问题，而这部分主要相关的是 DMA 直接内存访问，以及 IRQ 中断请求，只要解决这两个方面的隔离、保护以及性能问题，就是成功的 I/O 虚拟化。

3. 如何利用 VT-d 做到 I/O 的虚拟化

I/O 设备会产生非常多的中断请求，I/O 虚拟化必须正确地区分这些中断，将不同的中断提交到不同的 VM 上。传统设备的中断请求可以具有两种方式，第一种就是通过 I/O 中断总控器提交，另一种是通过 DMA 写请求直接发送出去。由于需要在 DMA 请求置入目标内存地址，因此这个架构需要完全访问所有的内存地址，并不能实现中断隔离。

VT- d 实现的中断重新映射（interrupt-remapping）架构通过重写消息中断（MSI）的格式来解决这个问题，新的 MSI 仍然是一个 DMA 可写请求的形式，不过并不嵌入目标内存地址，取而代之的是一个消息 ID，通过维护一个数据表，硬件可以通过不同的消息 ID 辨认不同的 VM 区域。VT-d 实现的中断映像可以支持所有的 I/O 来源，包括 IOAPICs，以及所有的中断类型，如一般的 MSI 以及扩展的 MSI-X。

VT-d 进行的变动还有很多，如硬件缓冲、地址翻译等，通过这些种种措施，VT-d 实现了北桥芯片级别的 I/O 设备虚拟化。VT-d 最终实现到虚拟化模型上的就是新添加了两种 I/O 虚拟化方式。

4. 从 CPU 到芯片组的硬件厂家支持

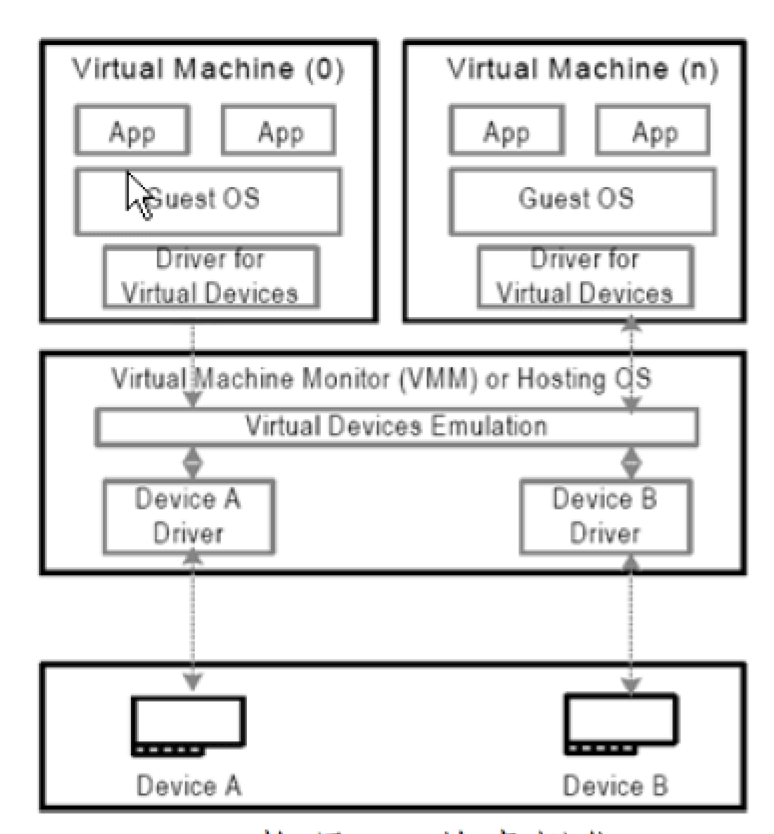
从 Intel 或 AMD 这两大 CPU 兼芯片组厂家的虚拟化技术发展路线图来看，虚拟化无疑是从处理器逐渐扩展到其他设备的，从 VT-i/VT-x 到 VT-d 就非常明显地体现了这个过程，从 CPU 一直到芯片组，都开始支持虚拟化的技术，对于着重 I/O 性能的企业级应用而言，落实了处理器的虚拟化和 I/O 的虚拟化，整个平台的虚拟化就贴近落实了，因此在未来，Intel 将会持续地开发 VT-d 技术，将各种 I/O 设备中加入虚拟化特性，从而提供一个强大的虚拟化基础硬件架构。

19.2.2 两种 I/O 虚拟化的方式

不管是 AMD 或是 Intel，其直接硬件虚拟化的原理都大同小异，主要分成传统的 I/O 类比虚拟化以及右边是直接 I/O 设备分配。在直接 I/O 设备分配中，虚拟机直接分配物理 I/O 设备给虚拟机，这个模型下，虚拟机内部的驱动程序和硬件设备直接沟通，只需要经过少量，甚至不经过 Hypervisor 的管理。

1. 物理 I/O 虚拟化

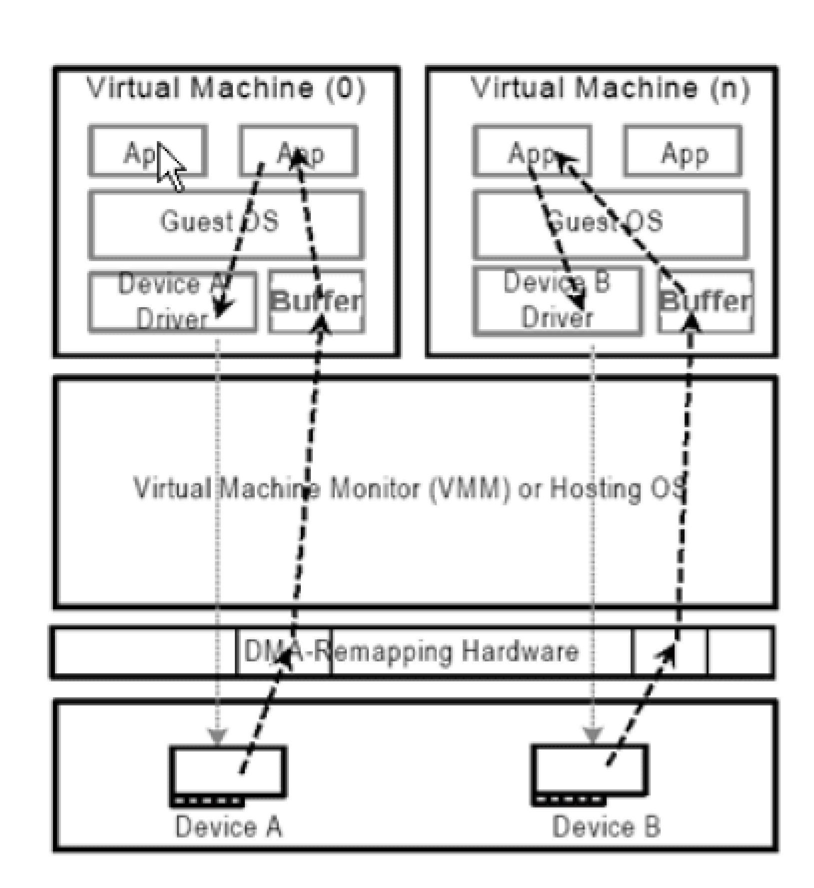
为了系统的稳定需要硬件的虚拟化支持，以隔离和保护硬件资源只给给定的 VM 使用，硬件同时还需要具备多个 I/O 分区来并发为多个 VM 服务，这个模型几乎完全消除了在 Hypervisor 中运行驱动程序的需求。比如 CPU，虽然 CPU 不算是一般认定的 I/O 设备，不过它确实就是通过这种方式分配给虚拟机，当然 CPU 的资源还处在 Hypervisor 的管理之下。



▲ 物理 I/O 的虚拟化

2. I/O 设备共享

第二种称为 I/O 设备共享，是 I/O 分配模型的一个扩展，对硬件具有很高的要求，需要设备支持多个功能通信端口，每个通信端口可以单独分配给一个 VM，这个模型无疑可以提供非常高的虚拟化性能体现。运用 VT-d 技术，虚拟机得以使用直接 I/O 设备分配方式或者 I/O 设备共享方式来代替传统的设备类比/额外设备通信端口方式，从而大大提升了虚拟化的 I/O 性能。



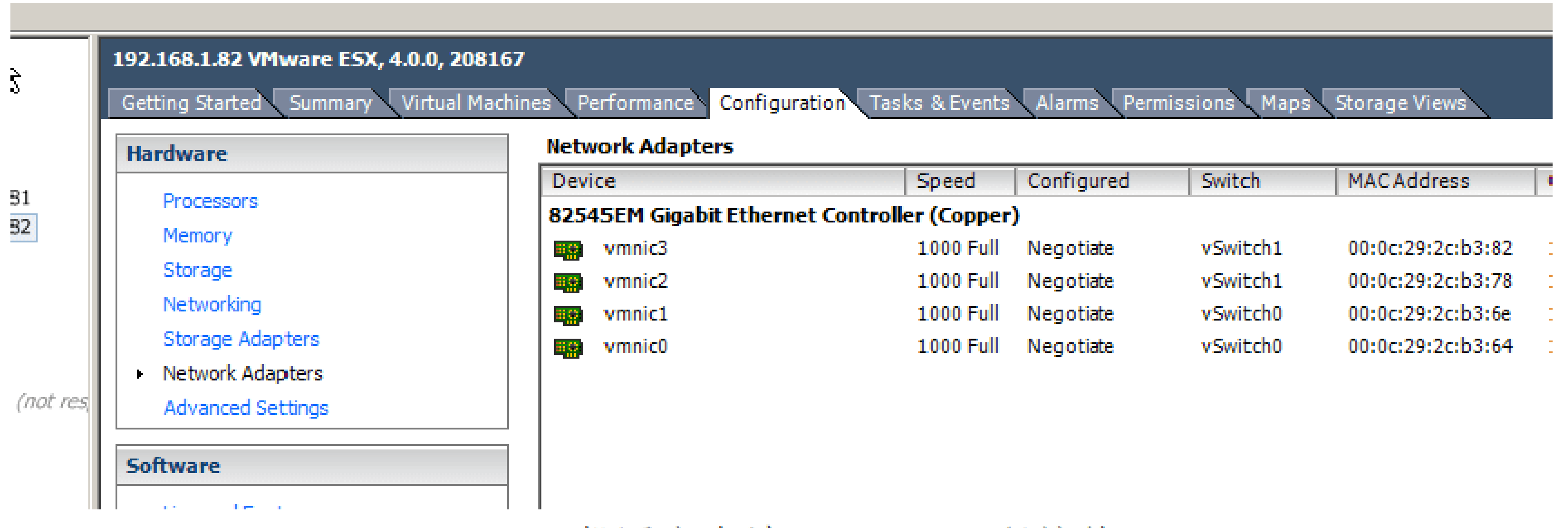
▲ 设备共享的结构

19.2.3 VMDirectPath I/O pass-through 的使用时机

vSphere 当然大量应用了 VT-d 及 AMD IOMMU 的特性，将直接硬件访问放入了产品中，因此只要你的 ESX/ESXi 硬件支持，就可以利用硬件功能落实直接硬件访问。在 vSphere 中，VM 直接访问硬件称为 VMDirectPath I/O pass-through 模式。事实上只要是在主板上支持了 VT-d 或是 IOMMU 功能，大部分的硬件都可以分出给 VM 使用。但在 vSphere 中，最常见的还是网卡和存储卡（HBA）两种。

1. 直接访问网卡资源

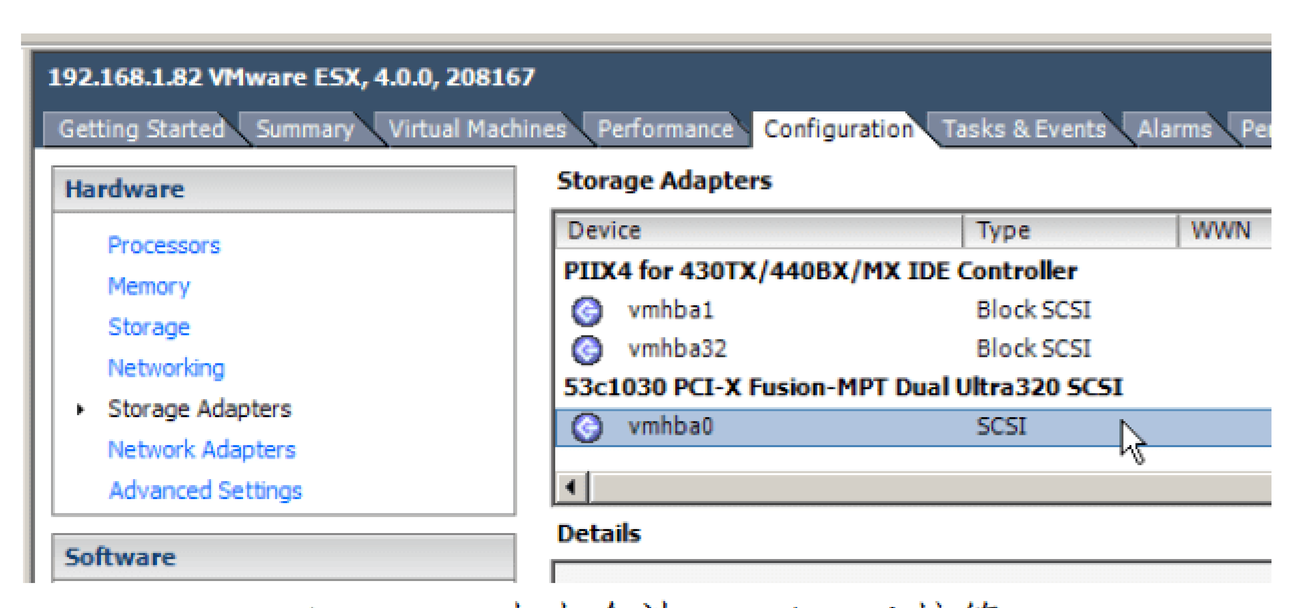
最常见的 VMDirectPath I/O pass-through 应用就是直接访问网卡。许多和网络安全相关的应用是需要直接访问网卡的，如 Switch 的 Port Mirroring，必须访问物理网卡，因此这一类的应用，如果需要转换到 VM 上，则需要 VM 直接访问网卡而非 Hypervisor 提供的虚拟网卡。



▲ 一般网卡会被 ESX/ESXi 所接管

2. 直接访问存储界面卡 HBA

虽然 vSphere 中的 VM 不支持直接访问本地硬盘的方式，但如果有 HBA 卡连接到外部的 LUN，那么也可以直接使用 VMDirectPath I/O pass-through 模式来访问外部的 LUN。这个方式和上一章介绍的 RDM 完全不同。RDM 模式还是要利用一个 VMDK 文件作为文件描述器，但使用 VMDirectPath I/O pass-through 连接至 HBA，再从 HBA 连接到外部的 LUN 时，完全不需要任何转换就可以访问 LUN，速度上不但比 RDM 模式更快，也能真正达到最高水平的兼容性。



▲ HBA 卡也会被 ESX/ESXi 接管

19.2.4 支持 VMDirectPath I/O pass-through 的前提

想要在 vSphere 中使用 VMDirectPath I/O pass-through，首先就是 CPU 要支持 Intel VT-x 或是 AMD 的 AMD-V。当 CPU 支持了上列的产品之后，接下来就是芯片组必须支持 Intel 的 VT-d 或是 AMD 的 IOMMU。

1. 支持 VT-x 的 CPU

下面是本书出版时，支持 Intel VT-x 的 CPU，但要注意的是，vSphere 必须使用 64 比特的 CPU，因此在使用以下的 CPU 时，必须确定其为 64 比特的。

▶ 支持 VT-x 的 Intel CPU

- Pentium 4662 及 672
- Pentium Extreme Edition 955 及 965 （not Pentium 4 Extreme Edition with HT）
- Pentium D 9x0
- Core 2 Duo E6300, E6400, E6320, E6420, E6540, E6550, E6600, E6700, E6750, E6850 （Conroe）
- Core 2 Duo E5400, E7600, E8200, E8300, E8400, E8500, E8600 及 some versions of the E7400 及 E7500 （Wolfdale）
- Mobile Core 2 Duo T5600, T6670, T7100, T7200, T7250, T7300, T7400, T7500, T7600G,

T7700，T7800，U7500，L7200，L7300，L7400，L7500，L7700，U7500，U7600，U7700（Merom）

- Mobile Core 2 Duo SU7300, SU9300, SU9400, SU9600, SL9300, SL9380, SL9400, SL9600, SP9300, SP9400, SP9600, P7350 （mac）,P7370, P7550 （confirmed）, P7570, P8400, P8600, P8700，P8800，P9500，P9600，P9700，T8100，T8300，T9300，T9400，T9500，T9550，T9600，T9800, T9900 （Penryn）
- Core 2 Quad Q6600, Q6700 （Kentsfield）
- Core 2 Quad Q8400，Q8400S，Q9300，Q9400，Q9400S，Q9450，Q9550，Q9550S，Q9650 及某些版本的 Q8300 （Yorkfield）
- Core 2 Extreme X6800 （Conroe XE）
- Core 2 Extreme QX6700, QX6800, QX6850 （Kentsfield XE）
- Core 2 Extreme QX9650, QX9770, QX9775 （Yorkfield XE）
- Atom Z520, Z530, Z540, Z550 （Silverthorne）
- Intel Core i3
- Intel Core i5
- Intel Core i7
- Pentium Dual-Core E6300, E6500, E6600、某些版本的 E5300 及 E5400
- Celeron SU2300, E3200, E3300, E3400
- 所有的 Xeon 3300 以上，5000, 7000 系列

2. 支持 VT-d 的芯片组

在确定 CPU 支持 VT-x 后，我们就给出支持 VT-d 的芯片组。

▶ 支持 Intel VT-d 的芯片组

- Intel Q35
- Intel Q45
- Intel X38
- Intel X48
- Intel X58

- Intel DQ35MP
- Intel DQ35JO
- Intel DQ35JO
- Intel DQ35MP
- Intel DX38BT
- Intel DX48BT2
- Intel DQ45CB
- Intel DQ45EK
- Intel DX58SO

- Intel 32x0

- Intel 3450
- Intel 54x0
- Intel 5500
- Intel 5520

3. 支持 AMD-V 的 AMD CPU

由于 AMD 的产品线没有 Intel 这么多，因此支持 AMD-V 的产品也集中在后期的 CPU。

▶ 支持 AMD-V 的 CPU

- Athlon 64 （“Orleans”）
- Athlon 64 X2 （“Windsor”）
- Athlon 64 FX （“Windsor”）
- Athlon 64 及 Athlon 64 X2（“F”或“G”stepping 于 AM2）
- Turion 64 X2
- Opteron [21]
- Opteron [22]
- Phenom
- Phenom II processors
- 所有的 Socket 939
- Sempron（Sable, Huron，Sargas 除外）
- 所有的 AMD Opteron CPUs（Barcelona 产品线以后）

4. 支持 AMD 的 IOMMU 的芯片组

下表是支持 AMD IOMMU 的芯片组。

▶ 支持 AMD IOMMU 的芯片组

- AMD 890FX
- AMD SR5690/SR5670

19.2.5 实践 VMDirectPath I/O pass-through 模式

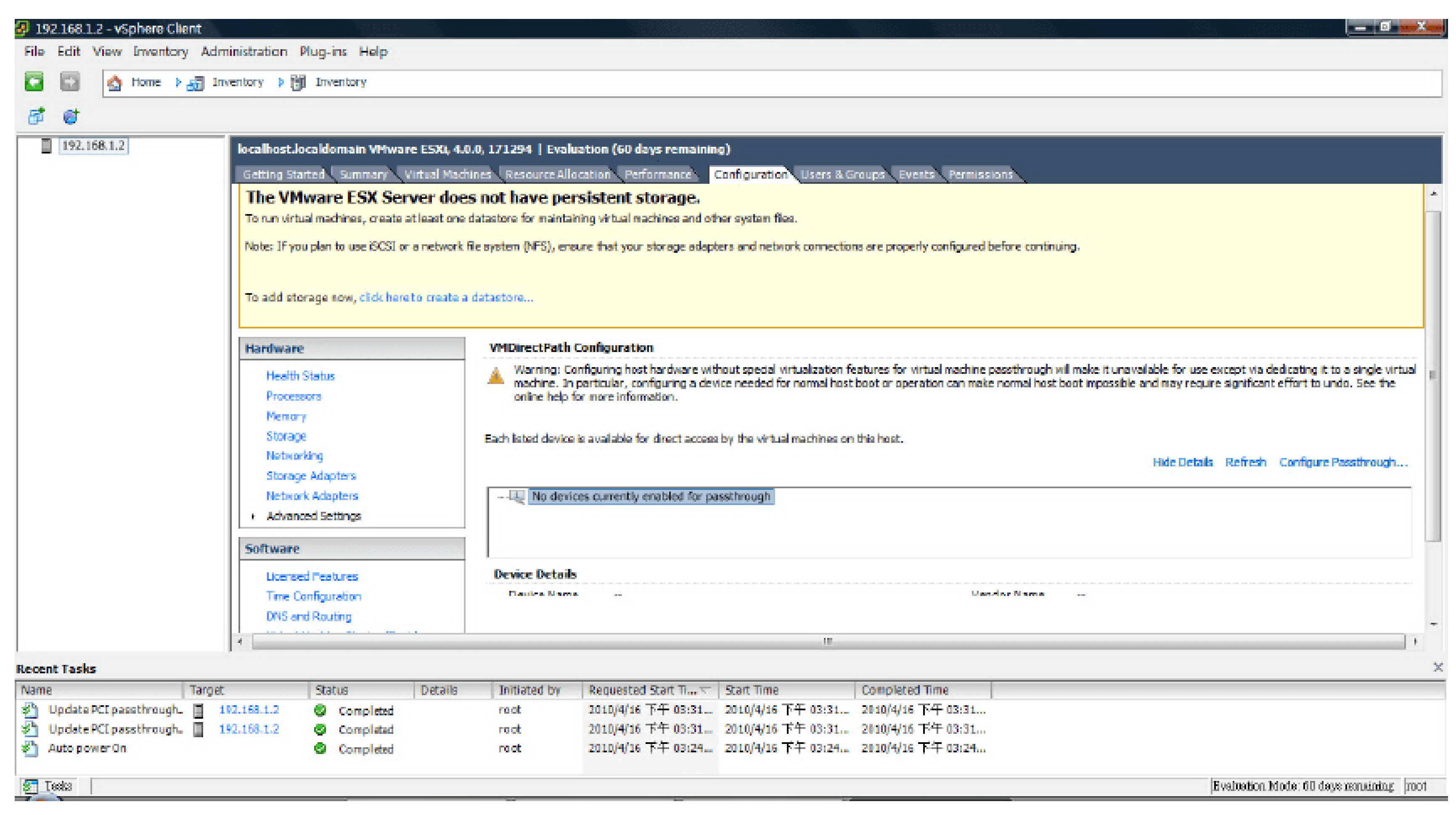
在 ESX/ESXi 中要实践 VMDirectPath I/O pass-through 相当简单，我们只要确定了 CPU 和芯片组支持 VMDirectPath I/O pass-through，只要在 vSphere 中将这个选项打开即可。

1. 激活 ESX/ESXi 下的 VMDirectPath I/O pass-through 硬件

在激活 VMDirectPath I/O pass-through 之前，我们要先确定在物理的硬件中，哪一个硬件要给 VM 使用。当确定之后，只要在 VMDirectPath I/O pass-through 的功能中给定即可。下面就是详细的步骤。

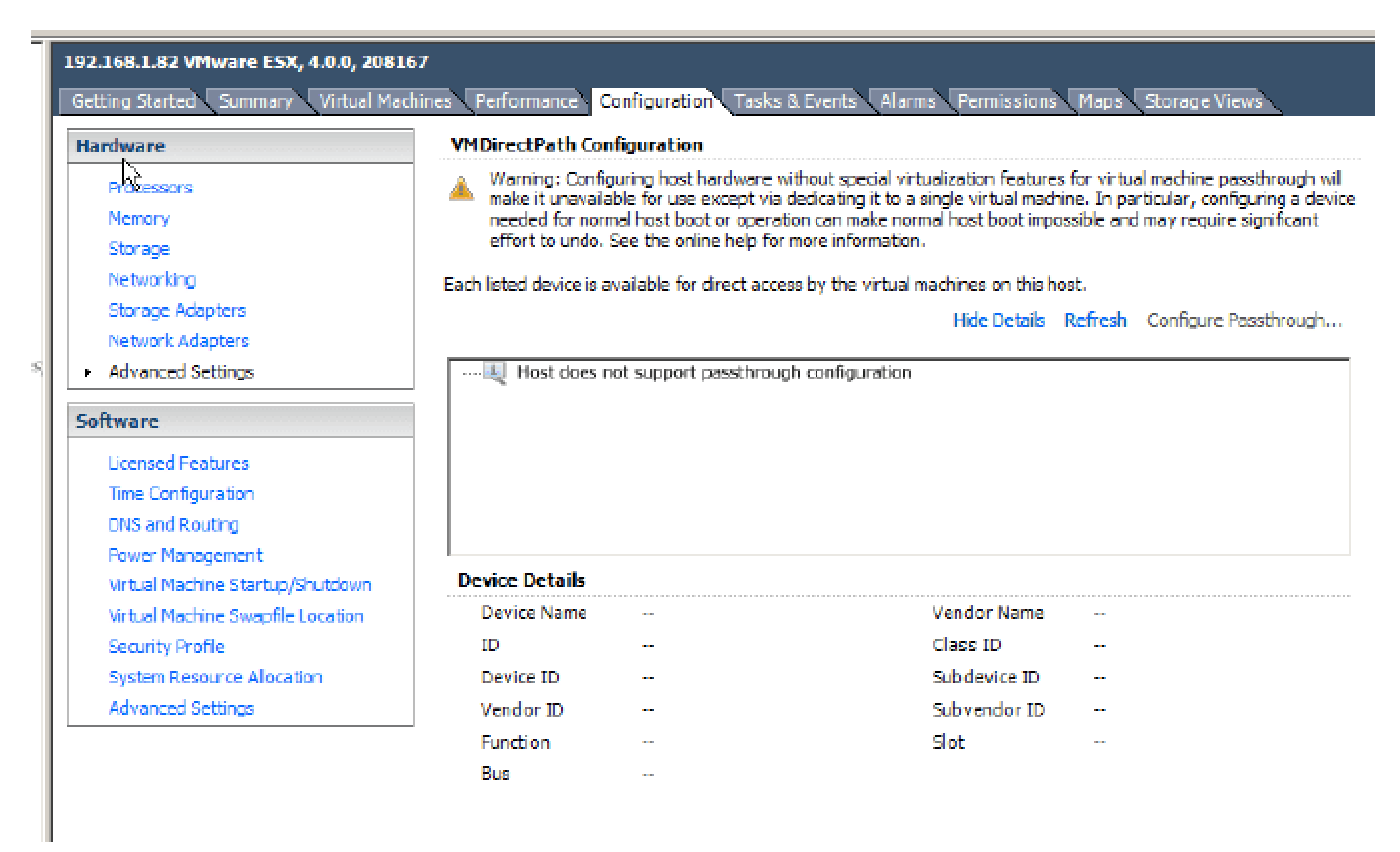
▶ 给定给 VM 用的 VMDirectPath I/O pass-through 硬件

1. 进入 vSphere Client，并且选择 Configuration/Hardware/Advanced Settings。此时我们会看到在窗口右边有一个 Configure Passthrough。



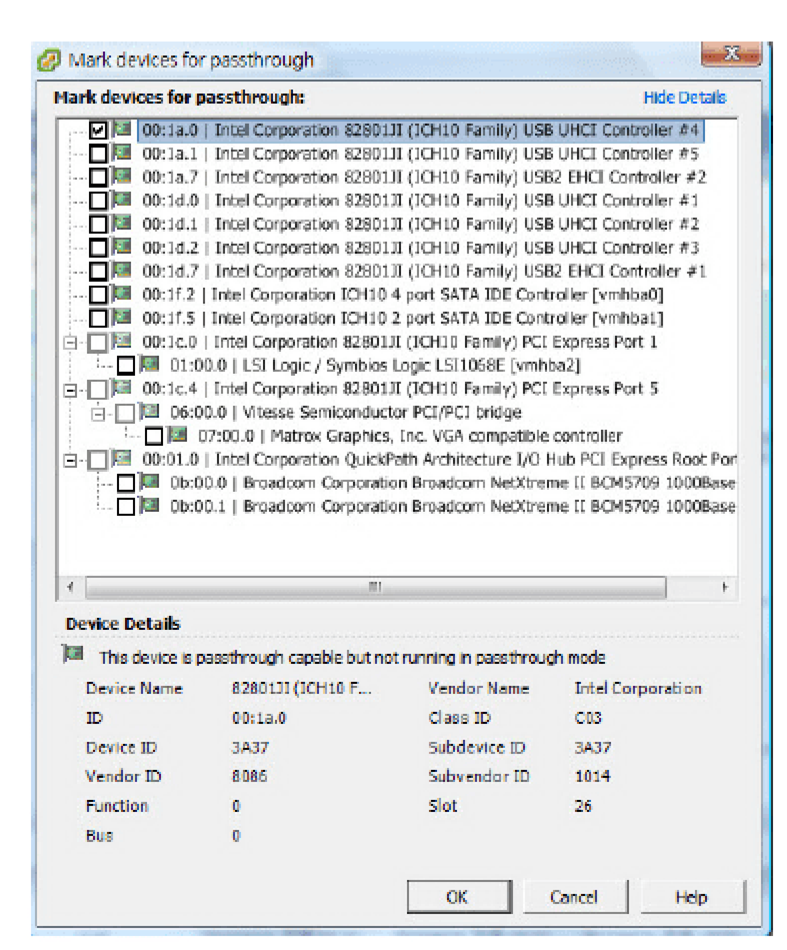
▲ 选择选项

2．如果你的硬件不支持 VMDirectPath I/O pass-through，则会弹出灰色画面。



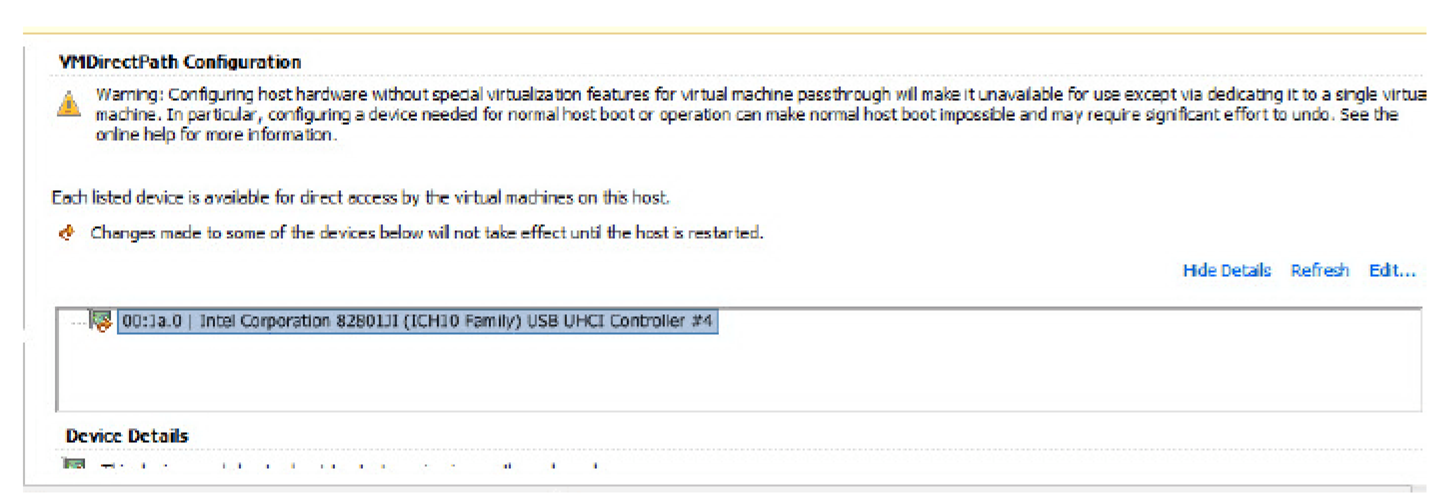
▲ 这是不支持的画面

3．进入之后，系统会给出可供 VMDirectPath I/O pass-through 的硬件。如果你要选择某个硬设备，选中即可。



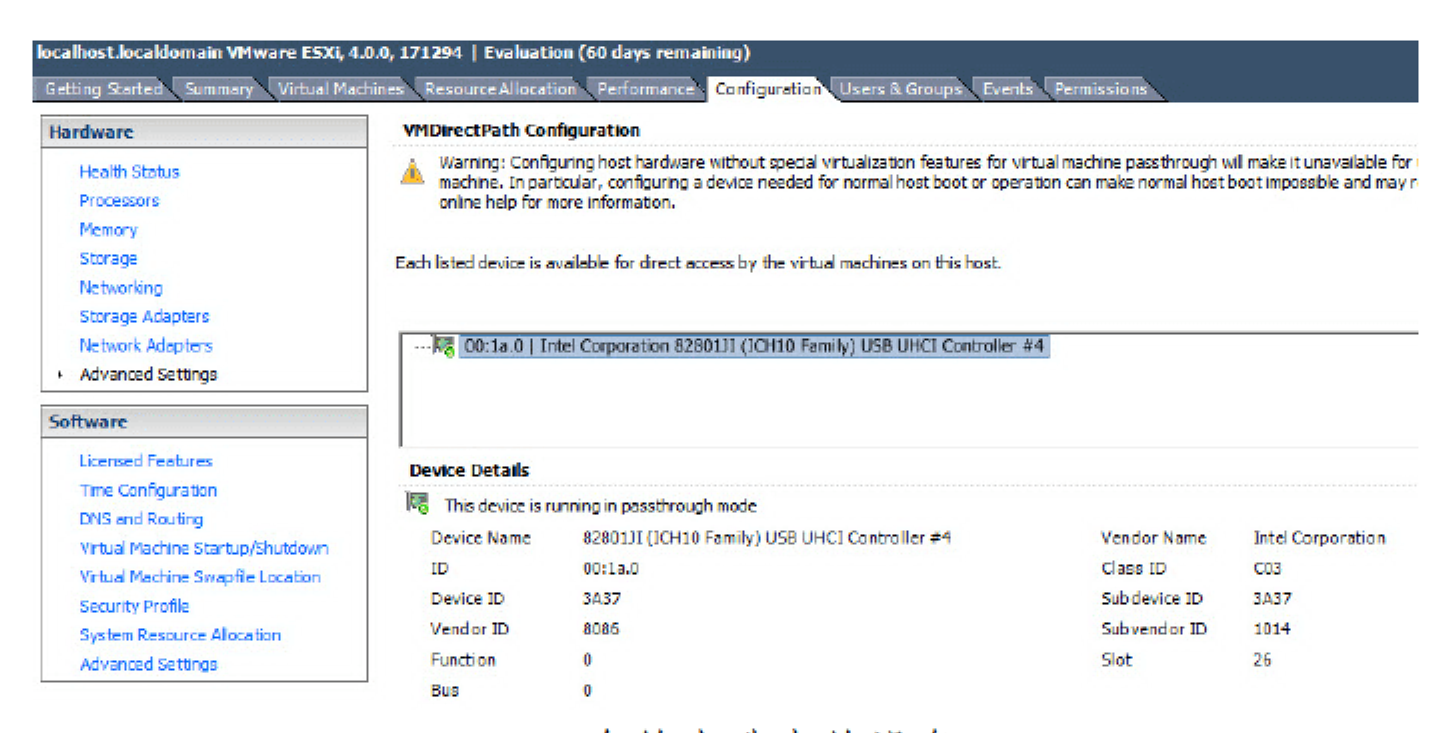
▲ 选中要给 VM 直接使用的硬设备

4．选中之后，你的 ESX/ESXi 必须重启，当然会弹出选择的硬件画面。



▲ 选择后，ESX/ESXi 会需要重启

5．接下来会回到 VMware Client 的画面，并且会给出选中的设备。此时这个设备，就可以被 VM 使用了。



▲ 会给出选中的设备

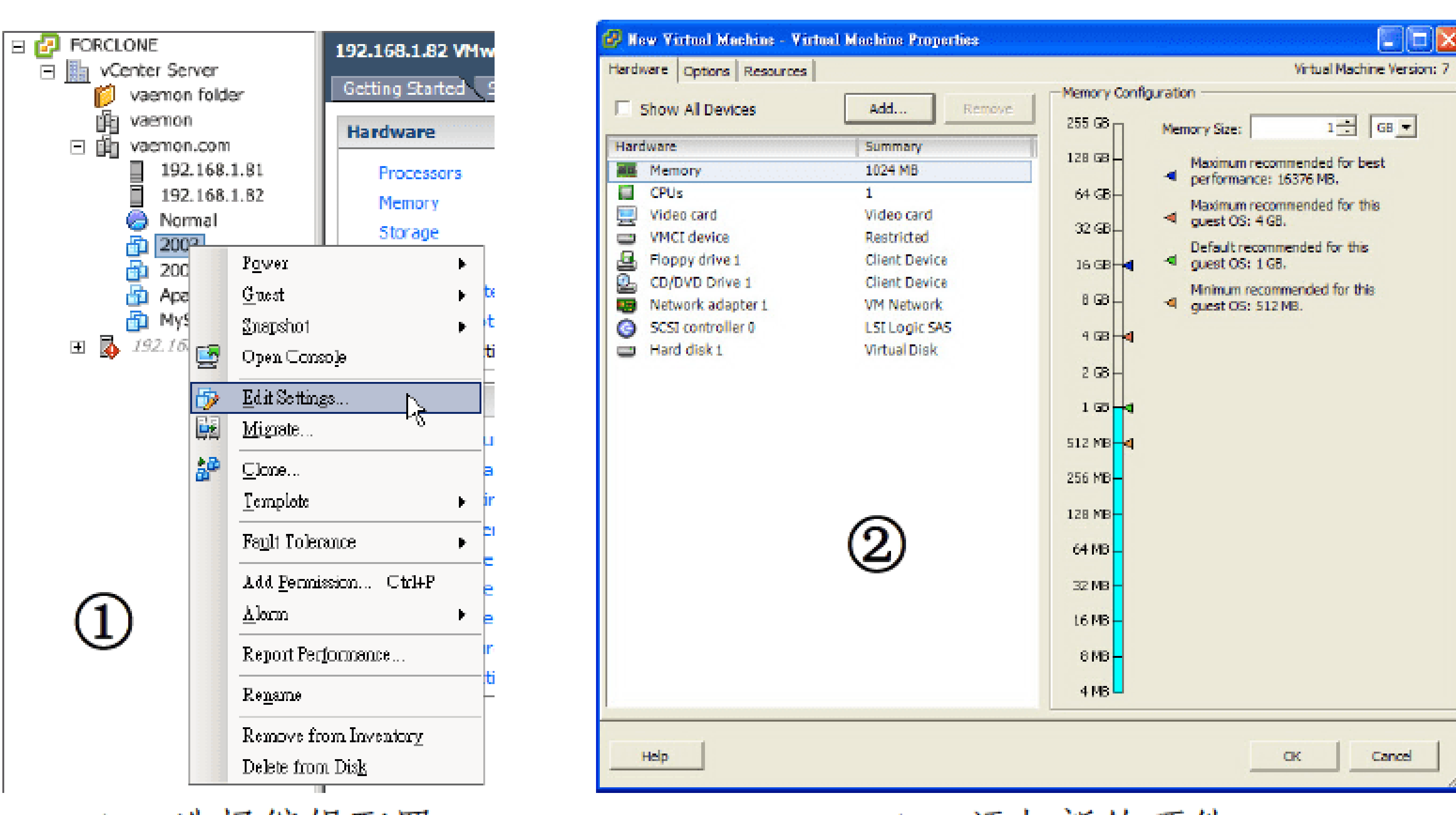
2. 在 VM 中使用 VMDirectPath I/O pass-through 的硬件

当我们在 ESX/ESXi 中给定了 VMDirectPath I/O pass-through 的硬件之后，在 VM 中就可以加入这个硬件了。下面就是详细的步骤。

▶ 将 VMDirectPath I/O pass-through 的硬件加入 VM 中

1．选择你要加入该硬件的 VM，并且选择 Edit Settings 选项。

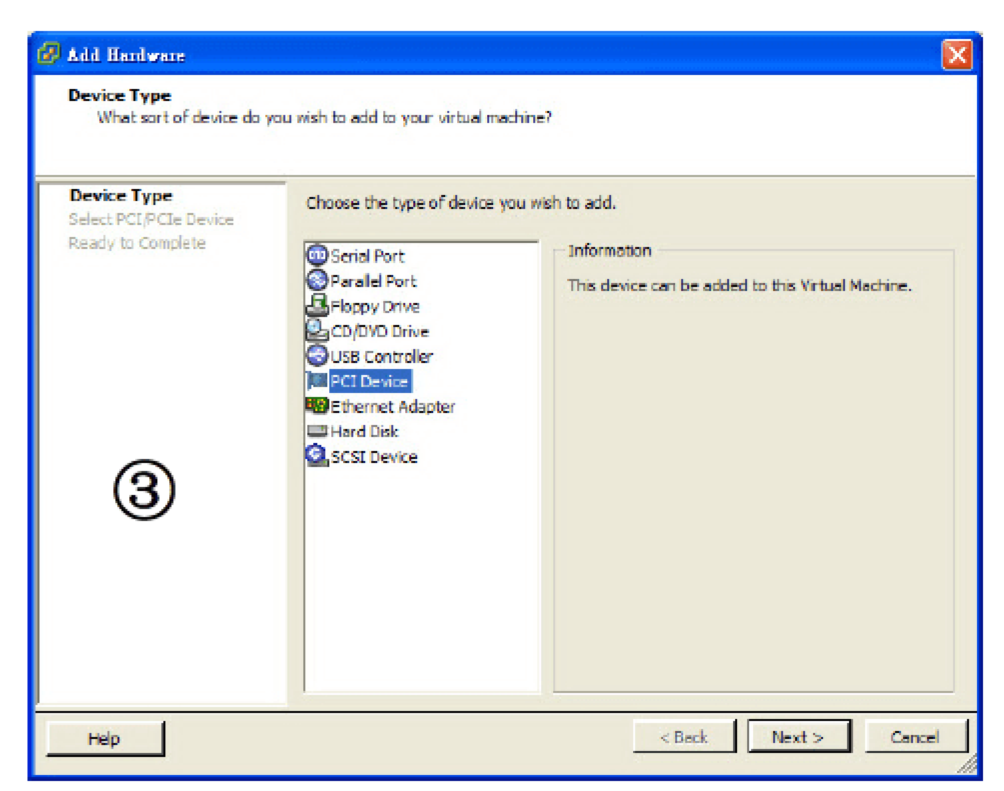
2．单击 Add 按钮。

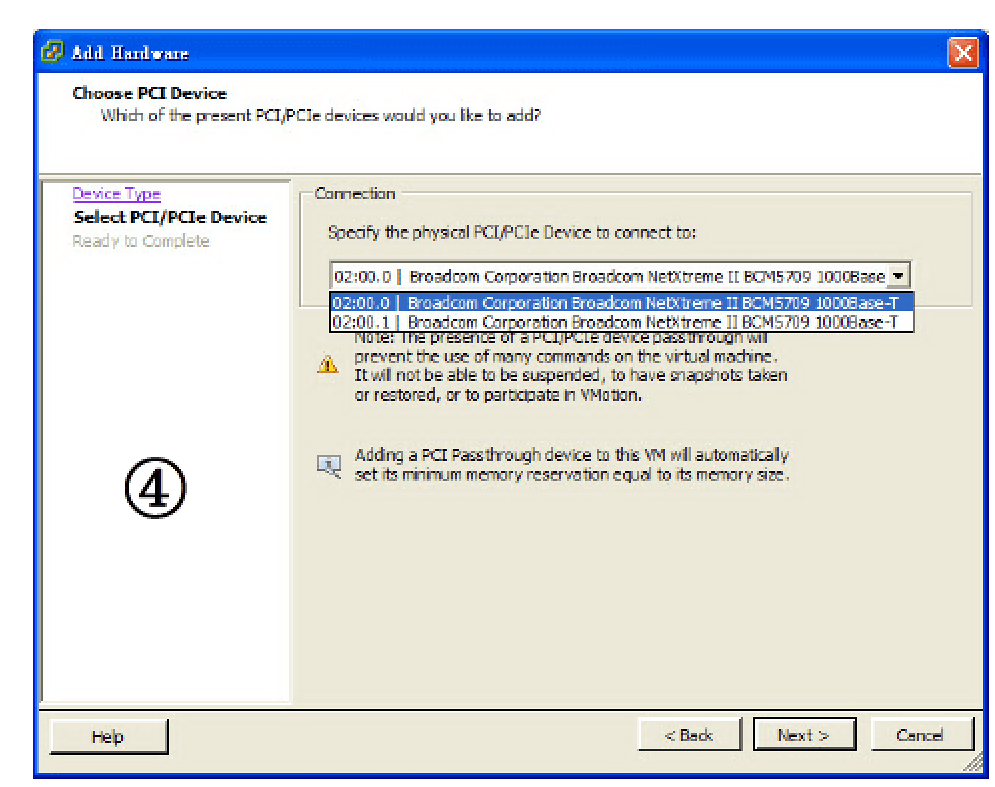


▲ 选择编辑配置　　　　▲ 添加新的硬件

3．选择 PCI Device 链接，之后单击 Next 按钮。

4. 此时会给出刚才在 VMDirectPath I/O pass-through 中选定的硬设备，如果你有多个设备，可以在菜单中选择。之后单击 Next 按钮。

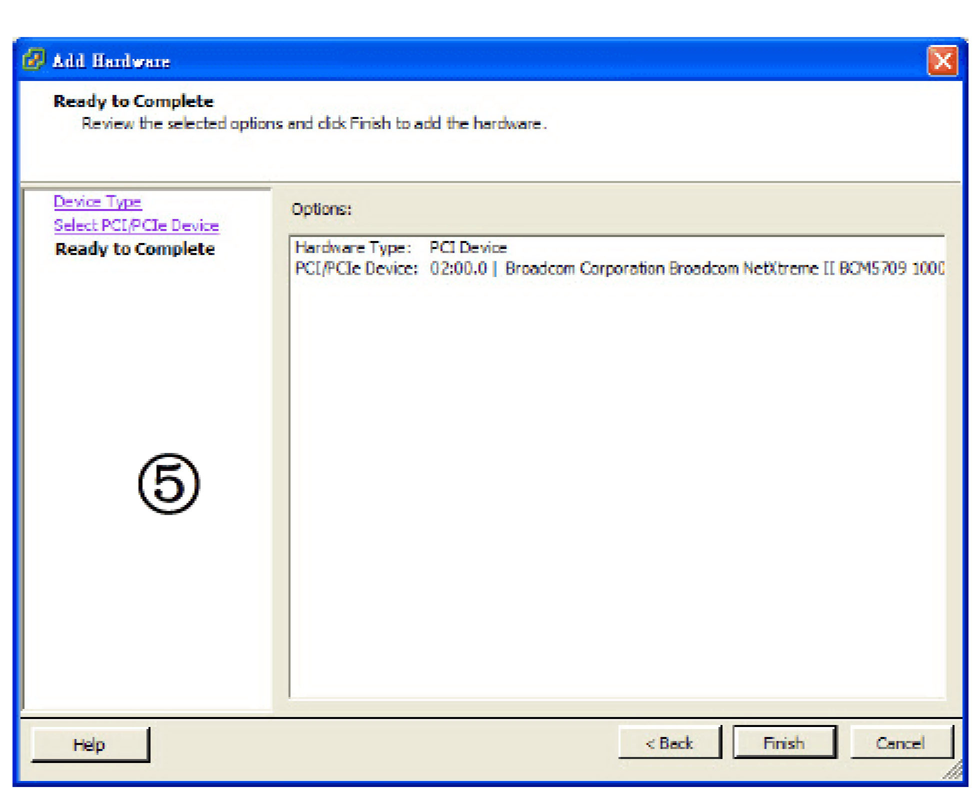


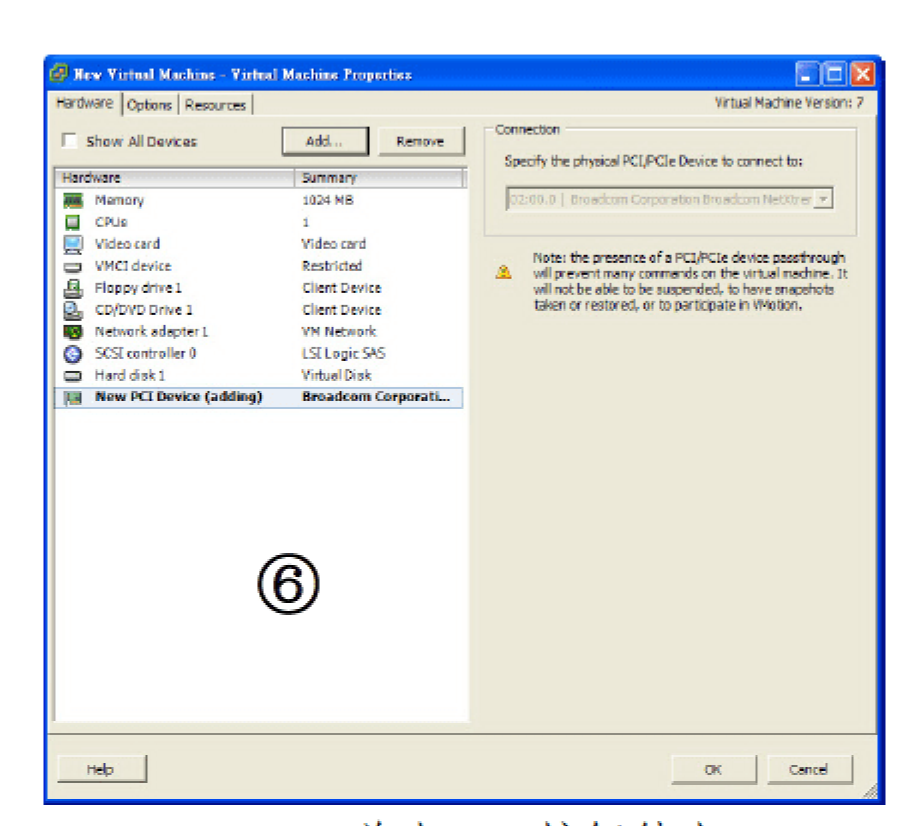


▲ 选择 PCI Device，所有的直接硬件都在此选择　　▲ 从中选择要直接使用的设备

5. 在弹出如下图所示的画面，意味着选择落实，我们在本例中选择了网卡。单击 Finish 按钮。

6. 接下来在硬件列表中会弹出加入的设备，单击 OK 按钮结束。

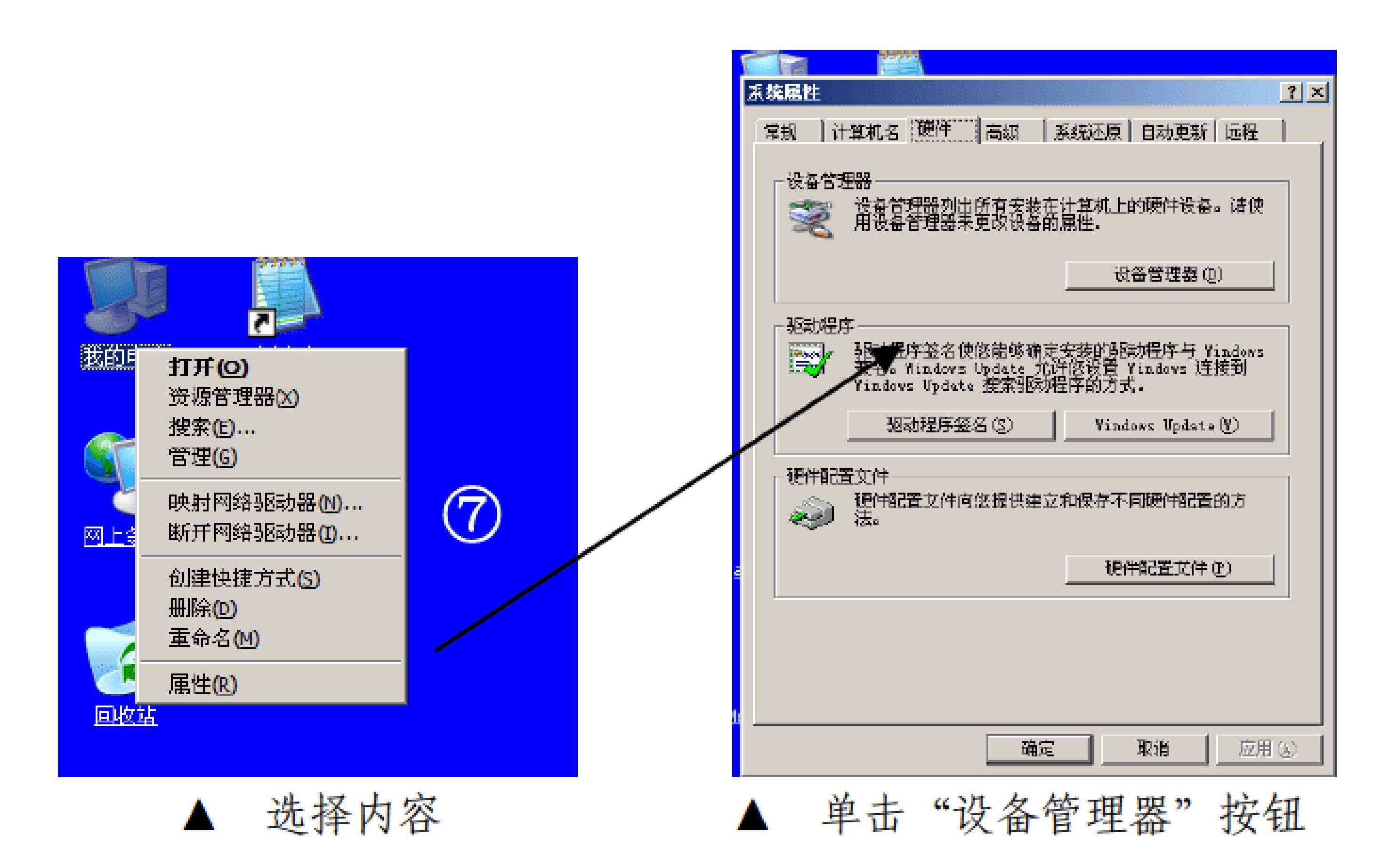






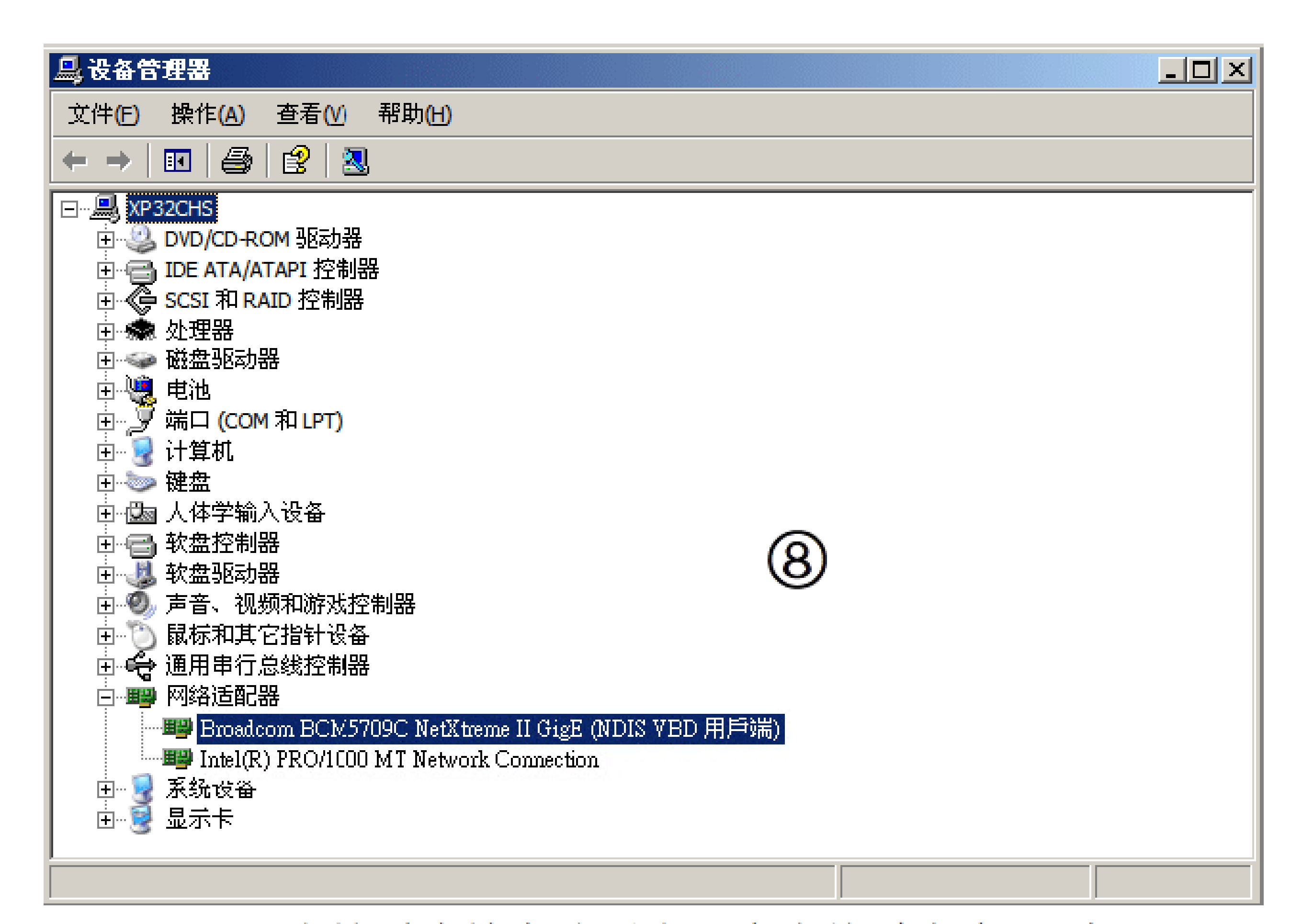
▲ 加入成功　　▲ 单击 OK 按钮结束

7. 接下来我们进入 VM。本例为 Windows 7，右击“我的电脑”。在弹出的下拉菜单中选择“属性”选项，然后在弹出的“系统属性”窗口中选择“硬件”选项卡，单击“设备管理器”按钮。



▲ 选择内容　　▲ 单击“设备管理器”按钮

8. 此时我们可以看到在 VM 中，可以完全抓到 VMDirectPath I/O pass-through 的硬件，为 Broadcom 的网卡。



▲ 这是刚才挑中的网卡，会直接弹出在 VM 中

如果你的硬件较为特殊，你还必须在 VM 中安装这个硬件的驱动程序。

结 语

直接硬件访问是 VM 中不可或缺的一个功能，在完美的世界中，所有的公司一开始就使用了 VM，但是在真实世界中，95%以上的公司在创建之初是以物理服务器架设公司平台的，因此一定会有物理到虚拟之间的转换。如果没有直接硬件访问，许多服务是无法上虚拟化平台的。Intel 在网络方面也将推出 VT-c，虚拟化的软硬件厂家不断推陈出新，让虚拟化更加方便，并发也提高了兼容性，RDM 和 VMDirectPath I/O pass-through 功能就是最好的实践。

第 20 章 使用 VCB 备份及还原 vSphere 上的虚拟机

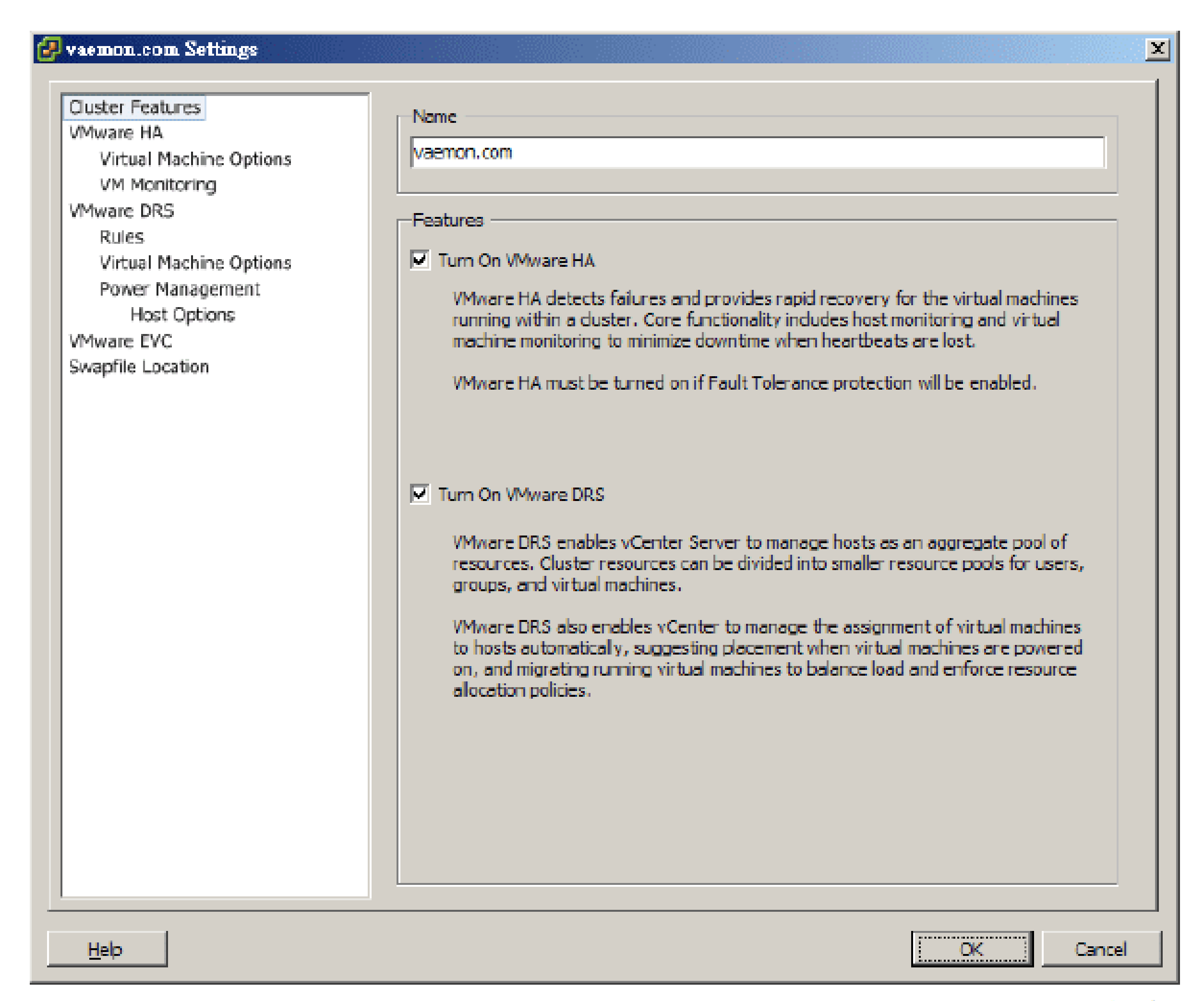
关键词：

- VCB 是 vSphere 入门版本的备份/还原软件
- VCB 不但本身是一个 Framework，也有提供命令进行备份还原
- VCB 在备份时，VM 可以引导
- VCB 在备份时，会先创建 VM 的快照，再将快照挂入备份
- VCB 的备份命令为 vcbVmName 导出备份参数
- VCB 的真正备份命令为 vcbMounter
- VCB 作为 Framework 时，可和第三方软件结合，使用 Symantec Backup Exec 2010 备份及还原 VM
- 学习利用 Symantec Backup Exec 2010 备份及还原 VM

在 vSphere 环境中的数据风险其实并不比物理机环境中小。由于一台服务器从前面的单个服务变成多个服务，因此存储设备上的数据丢失，会让多个服务并发中断。因此备份及还原的重要性绝对比从前的传统环境要高上很多。好在 VMware 的 VM 以文件为主，因此备份这些 VM 就是备份文件。VMware 提供入门版 vSphere VCB 产品，较高级版本的 vSphere 则可以使用 VDR，我们就在这一章先来看看 VCB 的使用。

20.1 备份及还原 VM 上的数据

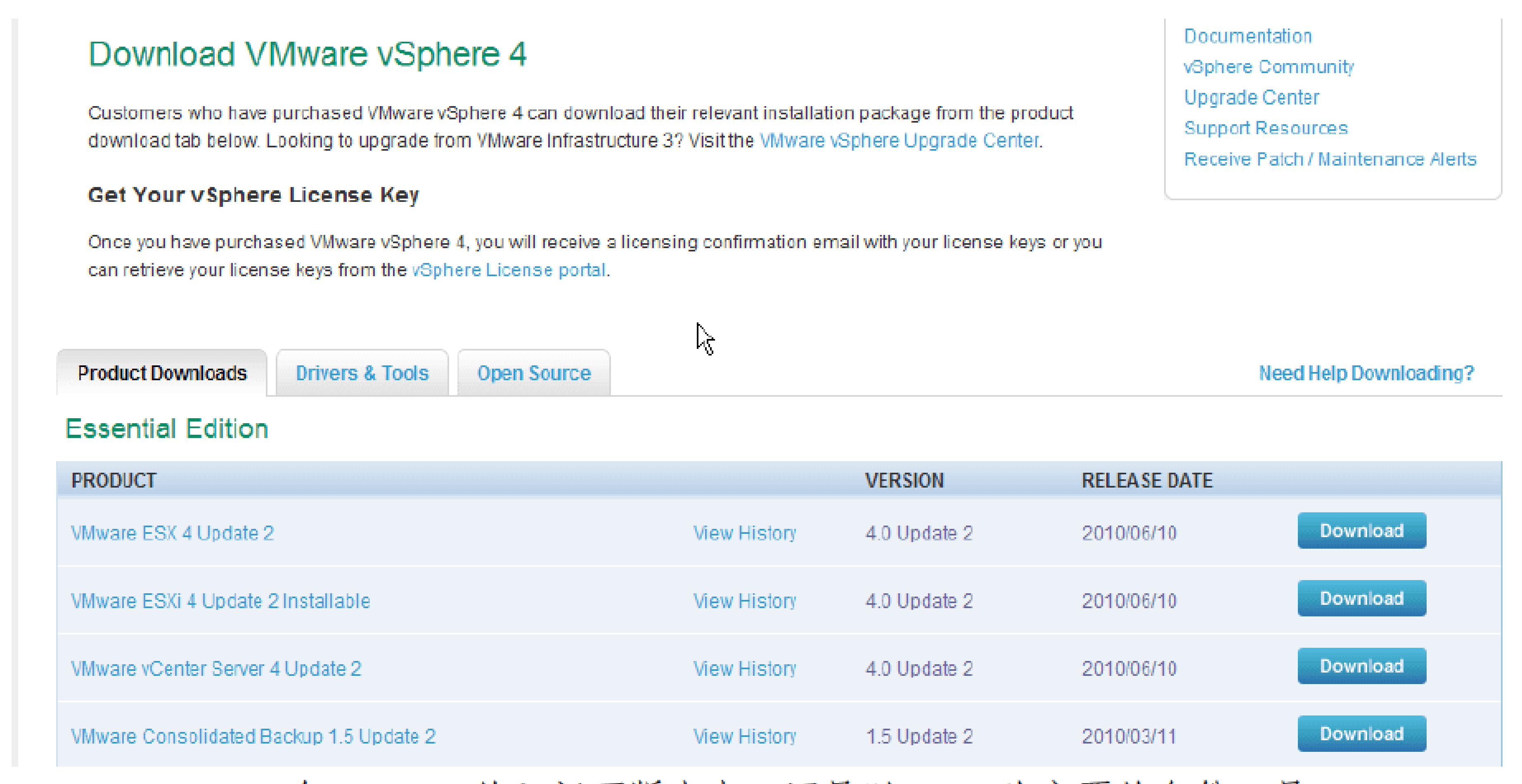
使用了 vSphere 之后，很多人觉得因为有 HA、DRS 或 VMotion 这些功能，系统的服务可以永远不中断，因此就认为备份不重要了。但这个观念却是非常危险的。VM 环境不但也需要备份和还原机制，更需要完善的计划和设备，因为 VM 环境的数据损失除了数据本身之外，操作系统、集群配置、快照、模板等都会不见，这些都是 vSphere 环境的最重要基础，因此 VMware 也针对整个虚拟架构提出了两大备份/还原架构，包括了 VMware Consolidated Backup（简称 VCB）以及 VMware Data Recovery（以下简称 VDR）。在本章的第一节，我们就先来看看 VCB。



▲ 有了 HA、DRS、VMotion，也无法保证数据不会消失

20.1.1 什么是 VCB

VCB 是 VMware 最具代表性的备份/还原机制。事实上在 VMware ESX 3.0 中就已经出现了。但 VCB 一直没有被大部分的用户注意，最主要的原因就是 VCB 的功能必须和第三方备份软件结合，如果要单独使用，必须通过难用的命令。但 VCB 在 vSphere 中仍然在入门的版本中提供。

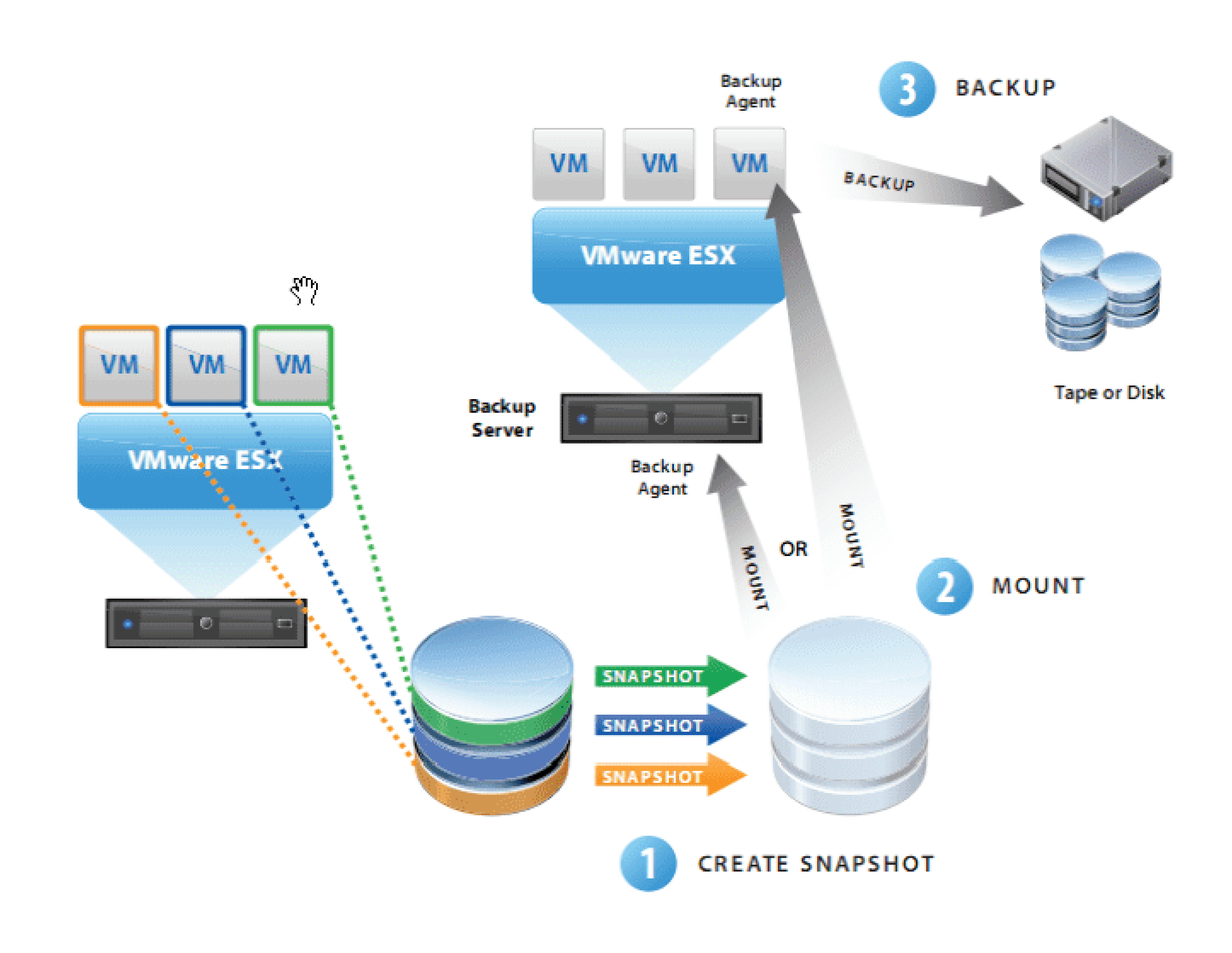


▲ 在 vSphere 的入门两版本中，还是以 VCB 为主要的备份工具

> **注意**
>
> **VCB 将走入历史**
>
> vSphere 推出之后，为了 VI3 的兼容性，VMware 在入门版本的 vSphere 还是建议使用 VCB，但在较高级的版本中，已经使用 VMware Data Recovery 来取代 VCB，而 VMware 也意味着，在以后推出的重大 vSphere 补丁版中，将主打 VDR，因此 VCB 在下一重大版本的 vSphere 中将不再支持，但 VCB 的兼容性及功能性，对简单环境的中小企业来说，一定比 VDR 好用很多，因此备份软件厂家还是不断在 VCB 的基础上加强自身的产品线。
>
> 
>
>
>
> ▲ VMware 将会在新一代的 vSphere 中不再提供 VCB

1. VCB 的角色及功能

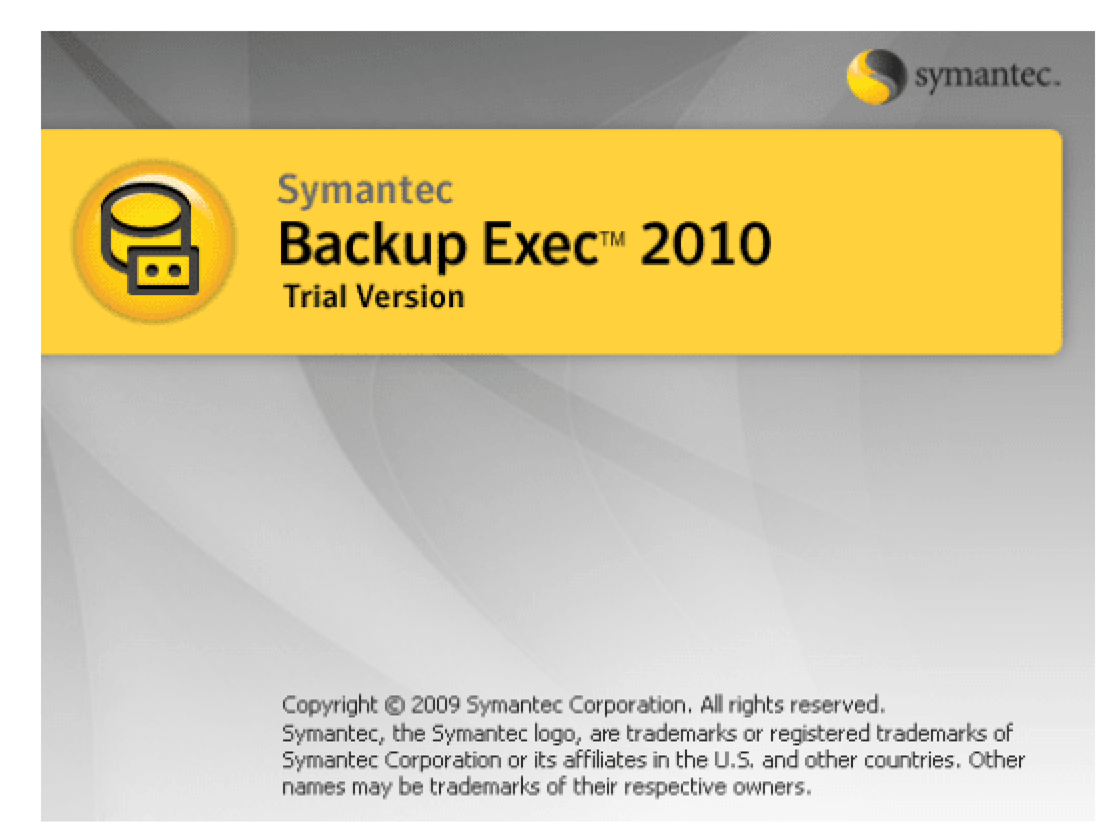
VCB 是一套单独的软件，必须安装在 Windows 2003/2008 上，当这台 Windows 2003/2008 安装了 VCB 之后，系统会将 VCB 的 Framework 及点命令安装起来，这台 Windows 2003/2008 就被称为 VCB Proxy Server。自此，不管是利用第三方软件通过 VCB Framework 或是使用 VCB 自带的点命令，都可以从这台 Proxy Server 连入 vCenter Server，并且备份 vCenter Server 上的所有 VM。



▲ VCB 的架构图

2. VCB 也是一个 Framework

VCB 本身也是一个 Framework，可以让第三方的备份软件调用，进行备份及还原的任务。事实上 VMware 的建议是将 VCB 视为一个 Framework，这样使用起来较为方便。当前主流的备份软件都能充分应用 VCB 这个 Framework，本章将会以最红的 Symantec Backup Exec 2010 为示例来介绍。



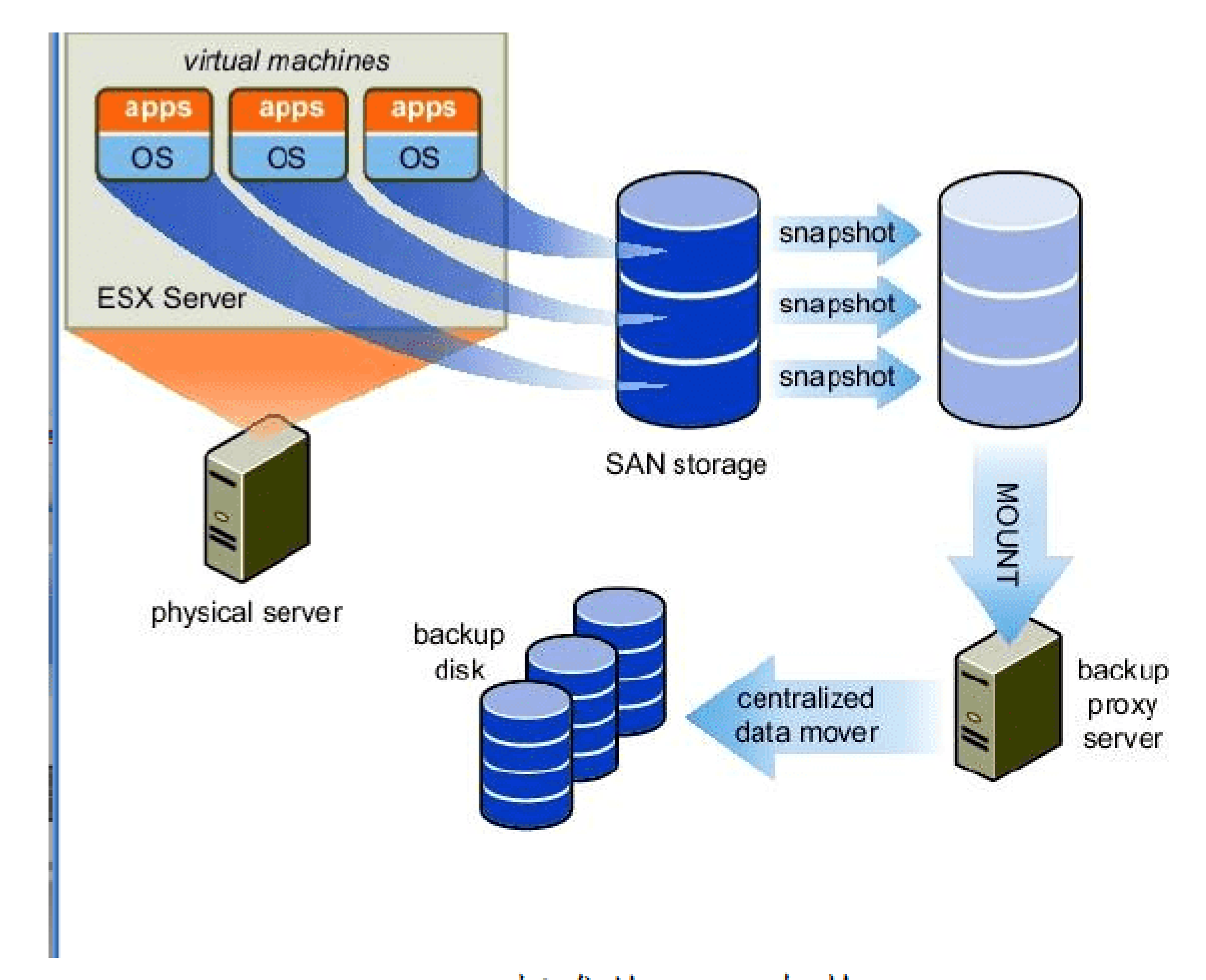
▲ Symantec Backup Exec 一直是 IT 人员最熟悉的备份软件

3. VCB 的部件及架构

VCB 可以备份 ESX/ESXi 单机，也可以备份 vCenter 中的所有主机，只要备份时给定就行。在一个 VCB 的架构中，必须有下列的几个主机。

4. 参加 VCB 的设备

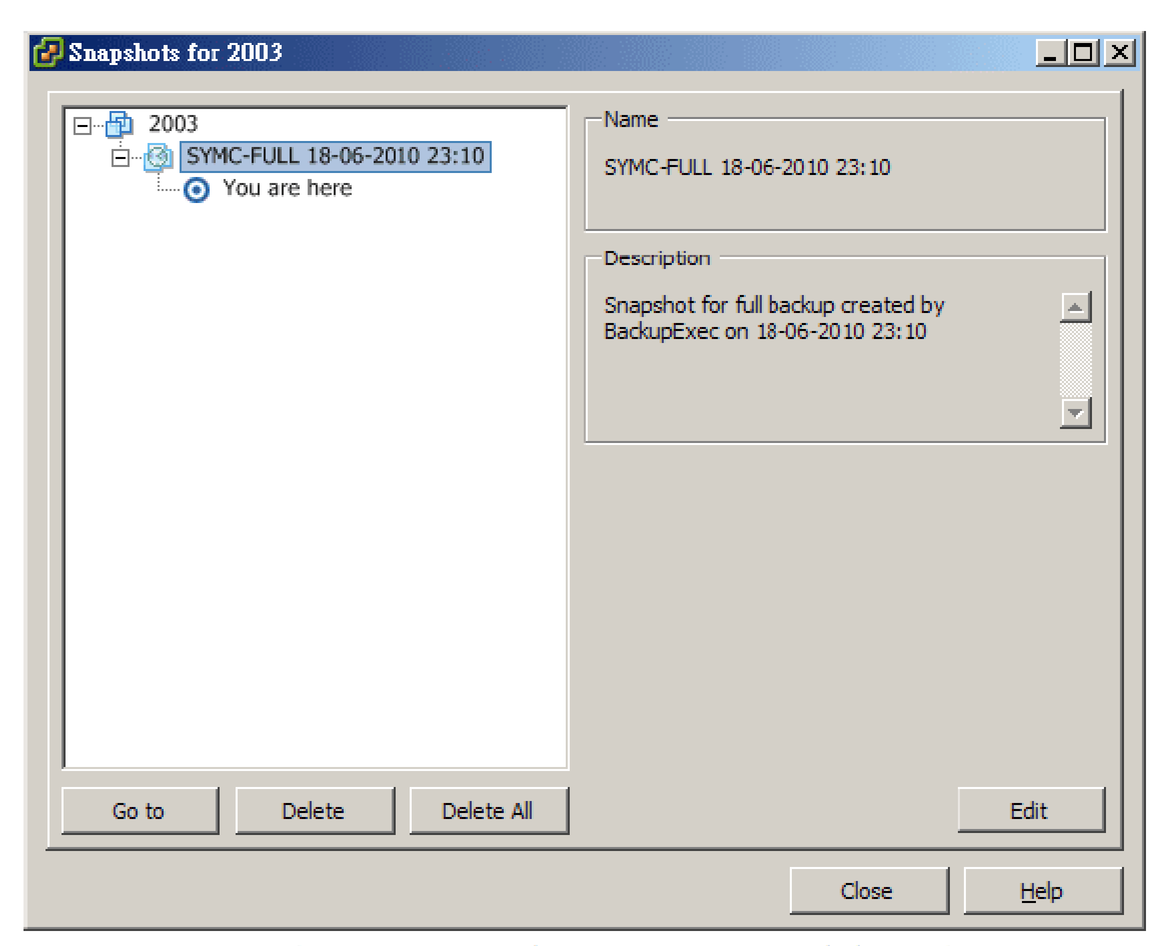
- Windows 2003/2008 的 Backup Proxy Server，安装 VCB Framework。
- 如果需要第三方备份软件，要安装在 Backup Proxy Server 上。
- 一台 ESX/ESXi 或 vCenter Server 管理多台 ESX/ESXi。
- 独立的存储介质。
- 所有参与备份的主机都要在同一个网段上。



▲ 标准的 VCB 架构

5. VCB 备份的原理

VCB 可以让 VM 在引导状态中进行备份，主要的方式就是使用最有名的快照。快照的使用方式在之前的章节都有提到。VCB 在进行备份之前，先将 VM 的磁盘创建快照，并且将快照装载 Backup Proxy Server 的磁盘，之后 VCB 再配合第三方软件将这个装载的磁盘备份起来，我们可以在 VMware 解释 VCB 的图中很清楚地看到这个过程。



▲ 在备份时，会发现系统会自动创建快照

20.1.2 在 vSphere 中备份 VM 的机制

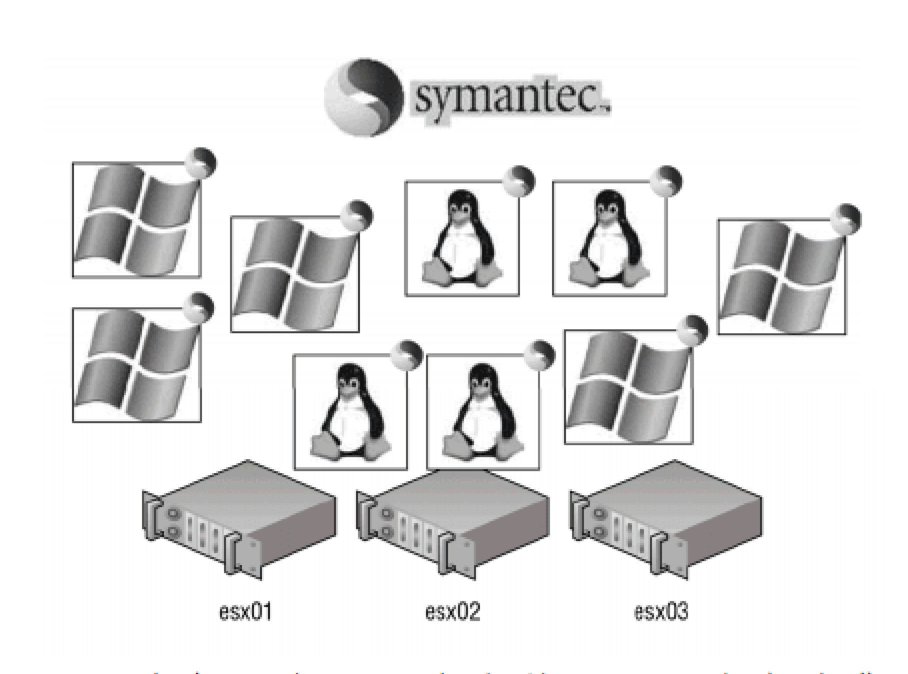
VM 和物理机一样需要备份，在物理机中我们会在需要的操作系统中安装备份软件，再将重要的文件备份到存储介质中。通常我们会将整个机器的操作系统先备份成映像档，而数据层则放入独立存储。由于 VM 和物理机的环境，数据层（如数据库或是网页数据），同样都放在独立存储中，因此两者的方式都大同小异，唯一的热点就是操作系统的备份了。

在 VM 中，备份的方式和物理机一样，但 VMware 提供了 VCB 让 VM 的备份更加弹性化以及整体化。我们就来看看在 vSphere 环境中的备份方式。

1. 在 VM 中使用传统的 Backup Agent

在物理机环境中，我们会在每一台操作系统中安装备份代理软件（Backup Agent）。在 VM 中也可以这么做，只要在每一台 VM 中安装备份代理软件，就可以备份 VM 中的所有数据。但这么做的坏处当然就是备份代理软件的授权费用，因为每一个 VM 需要一个代理软件，不管这台物理服务器的合并率是多少。

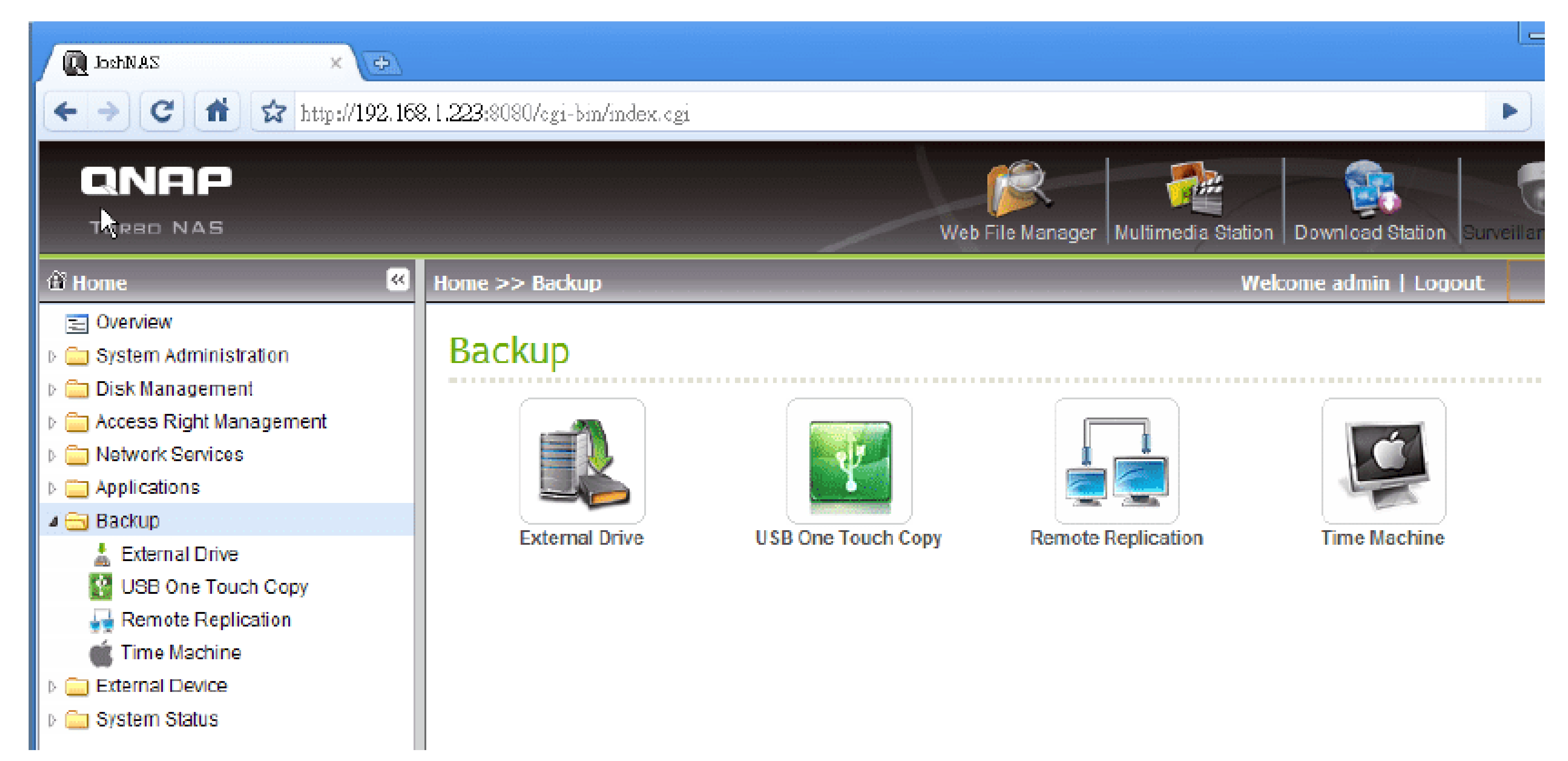
此外这么做的问题就是无法利用到 vSphere 的特殊功能，如果我们在企业提供服务的环境中都已经使用了 vSphere 的 HA、DRS、VMotion 等功能，但备份还原上还使用传统的方式，会无法和整个 IT 环境紧密结合。



▲ 在每一个 VM 中安装 Agent 有点浪费虚拟架构的优点了

2. 使用 SAN 本身的备份机制

vSphere 一定是运行在独立存储环境，因此不管是 SAN、NAS 或是 iSCSI，本身一定拥有自身的备份机制。事实上笔者是十分建议使用 SAN 的备份机制，只是在 VM 引导的情况下，会起到文件死锁的问题而无法备份，还好当前较新的存储设备都能嗅探到 VM 的存在，可以在 VM 引导时进行备份。



▲ 一般的 SAN 或 NAS 本身都有备份的机制

3. 使用 VCB 备份整个 VM

使用 VCB 备份整个 VM 的方式很简单，你可以直接使用第三方备份软件，也可以使用 VCB 的命令。我们会在本章稍后介绍完整的备份命令。

4. 使用 VCB 备份 VMDK 文件

你也可以使用 VCB 的命令来备份 VMDK 文件而已。事实上 VM 中最重要的就是 VMDK。由于 VMDK 是真正存放数据的地方，VMX 虚拟机组态档还可以重新制作，只是当 VM 越来越多时，VMX 组态档要重做也十分耗时，因此笔者还是建议要备份完整的 VM。

5. 使用第三方软件备份

本书的热点将会放在第三方软件备份。使用第三方软件的好处就是方便操作。VCB 最大的问题是没有提供一个 GUI 界面，对于习惯 Windows 界面的广大用户来说，第三方软件充分利用了 VCB Framework，再加上自身的界面，让大家在不改变使用习惯的情况下备份 VM。第三方软件也能选择完整的 VM、VMDK，或是以增量备份的方式来将 VM 改变的部分备份起来。



▲ 使用第三方备份软件也可以备份到 VMDK 中的文件，但要装 Agent

20.2 使用 VCB 命令备份及还原 VM

使用 VCB 备份 VM 首先就是要找一台 Windows 2003/2008 的服务器来安装 VCB 这套软件+Framework。VCB 可以在 vCenter Server 的安装光盘中找到，也可以去 VMware 的网站下载，当安装完毕之后，就可以开始使用了。我们就来看看。

> **注意**
> **如果你要使用第三方备份软件**
> 如果你使用第三方备份软件，最好先安装该软件再安装 VCB，因为 VCB 会自动找到当前主流的备份软件，并且将需要的部件和备份软件结合。如果你只要单独使用 VCB，则直接安装即可。

20.2.1 安装及使用 VCB

VCB 是一个独立的软件，必须安装在能连上 vCenter 的 Windows 2003 计算机上，我们就来看看如何安装及使用。

1. 安装 VCB

安装 VCB 就像安装任何 Windows 下的软件一样。

▶ 安装 VCB 到 Windows 2003 Server

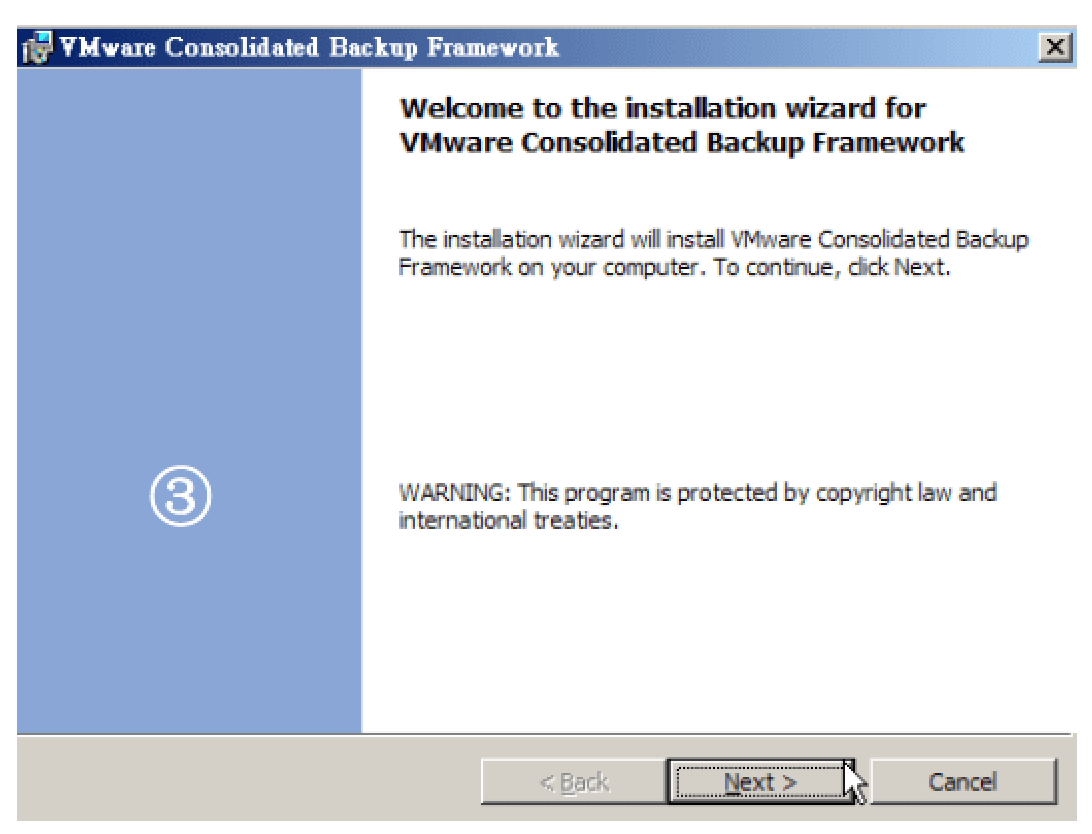
1. 打开 vCenter Server 的光盘片，并且在 vcb 目录下找到 VMware-vcb-150805.exe。直接运行这个文件。
2. 开始安装 VCB。



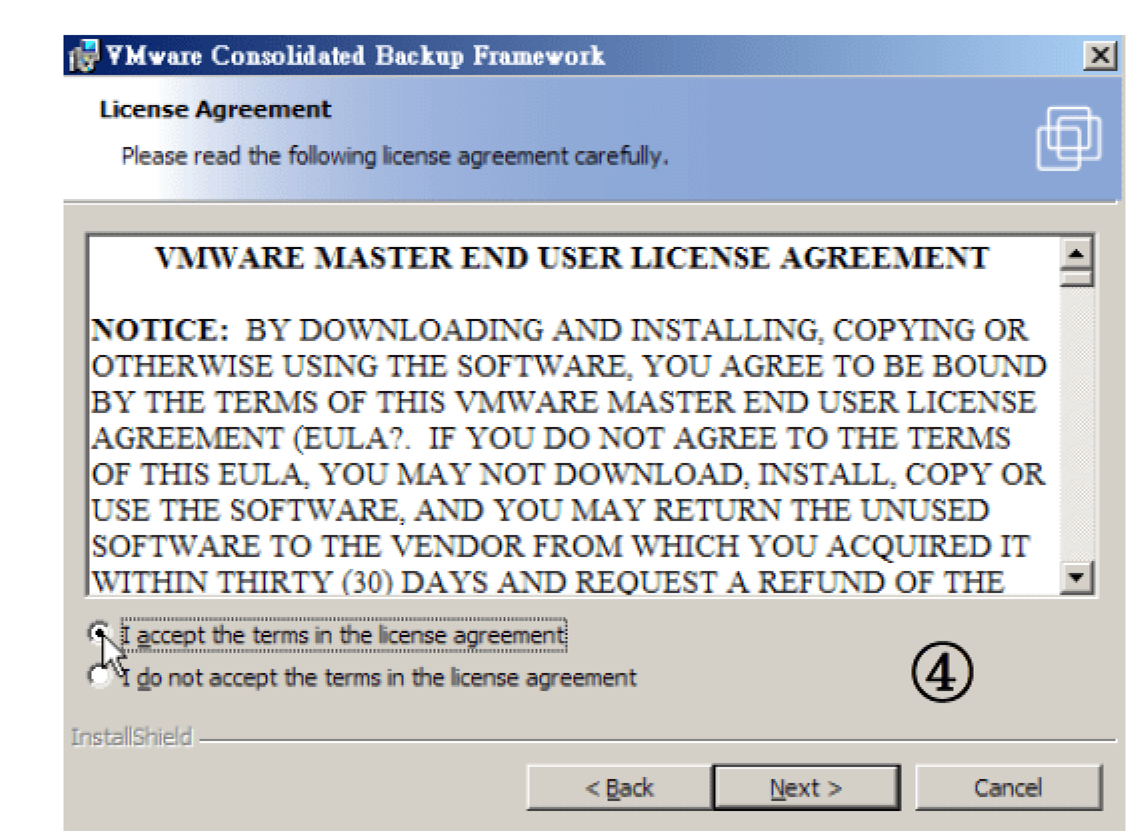


▲ 运行这个文件　　▲ 开始安装 VCB

3. 首先会弹出安装 VCB Framework 的画面，单击 Next 按钮继续。
4. 接着弹出授权同意，选择第一项接受，单击 Next 按钮继续。



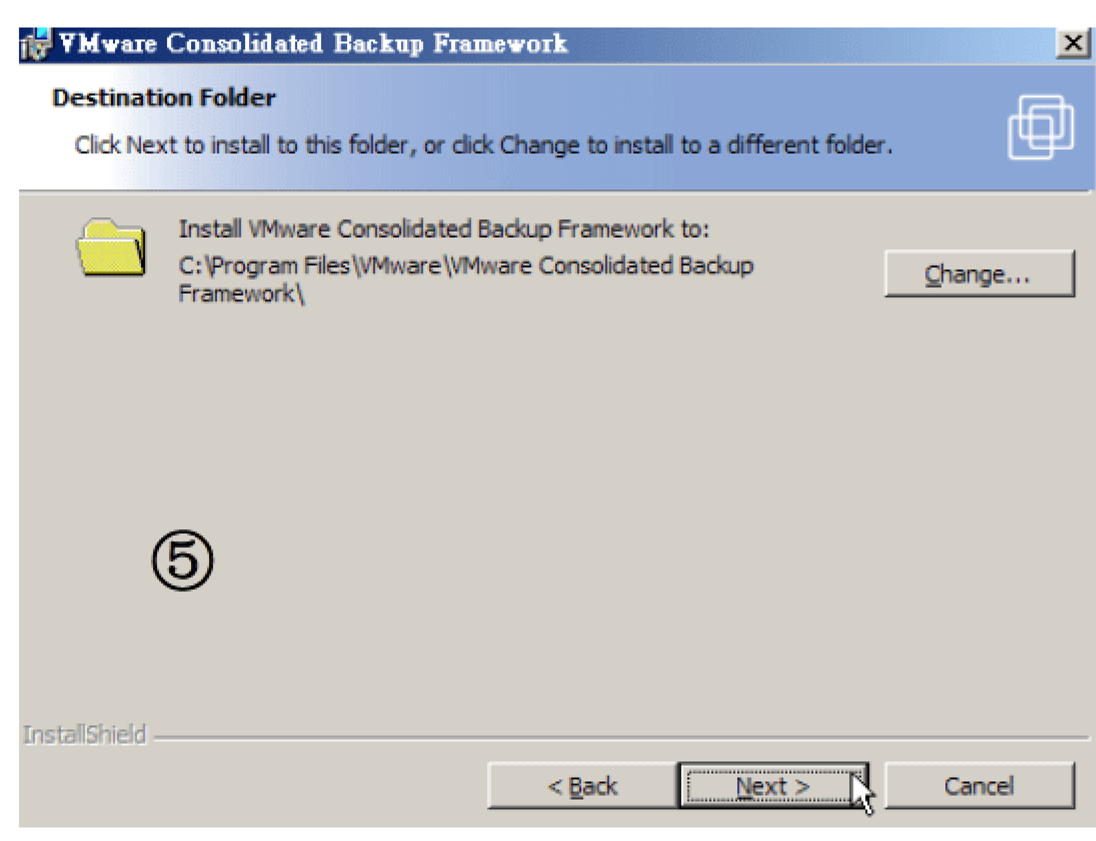
▲ 安装 VCB 的 Framework



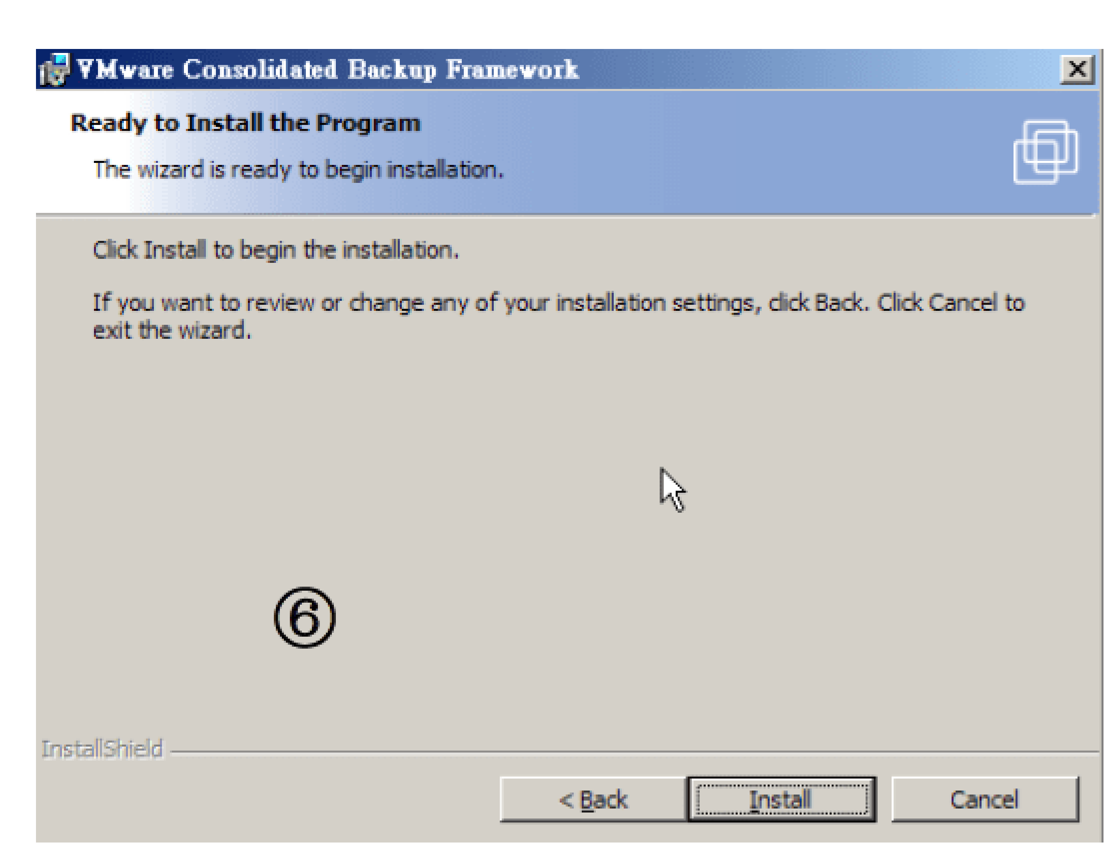
▲ 授权画面

5. 接下来是安装目录，要注意这个目录，因为以后会使用这个目录中的命令。单击 Next 按钮继续。

6. 接下来是总结画面，单击 Install 按钮开始安装。



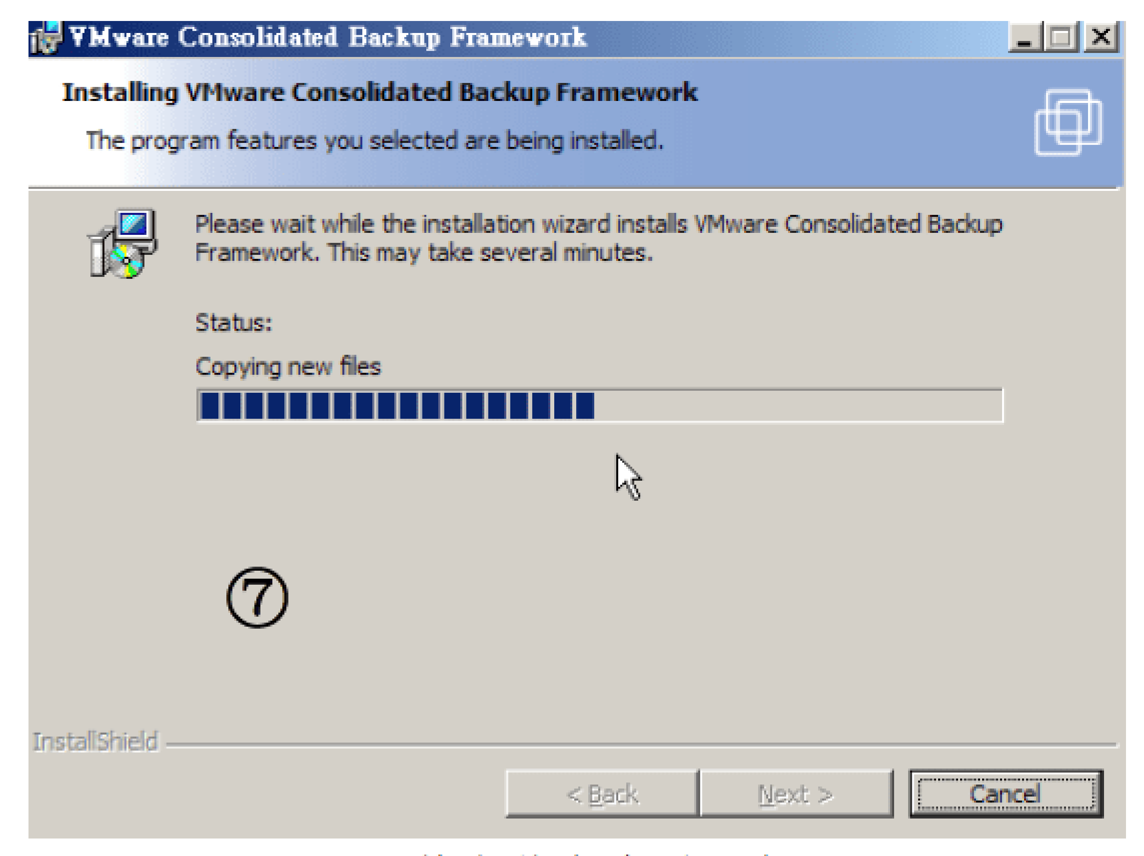
▲ 安装至目录，以后运行 VCB 命令都要在这个目录下



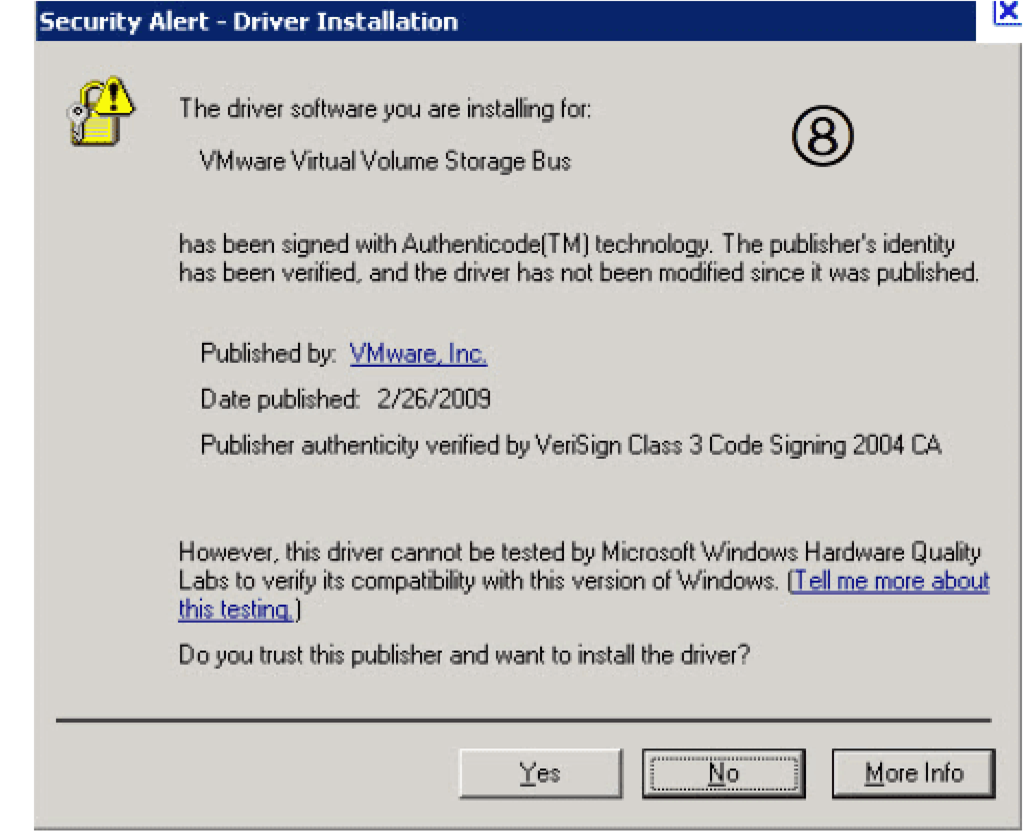
▲ 开始安装

7. 接下来开始安装。

8. 当弹出画面时，意味着要安装存储 Bus 的驱动程序，单击 Yes 按钮。



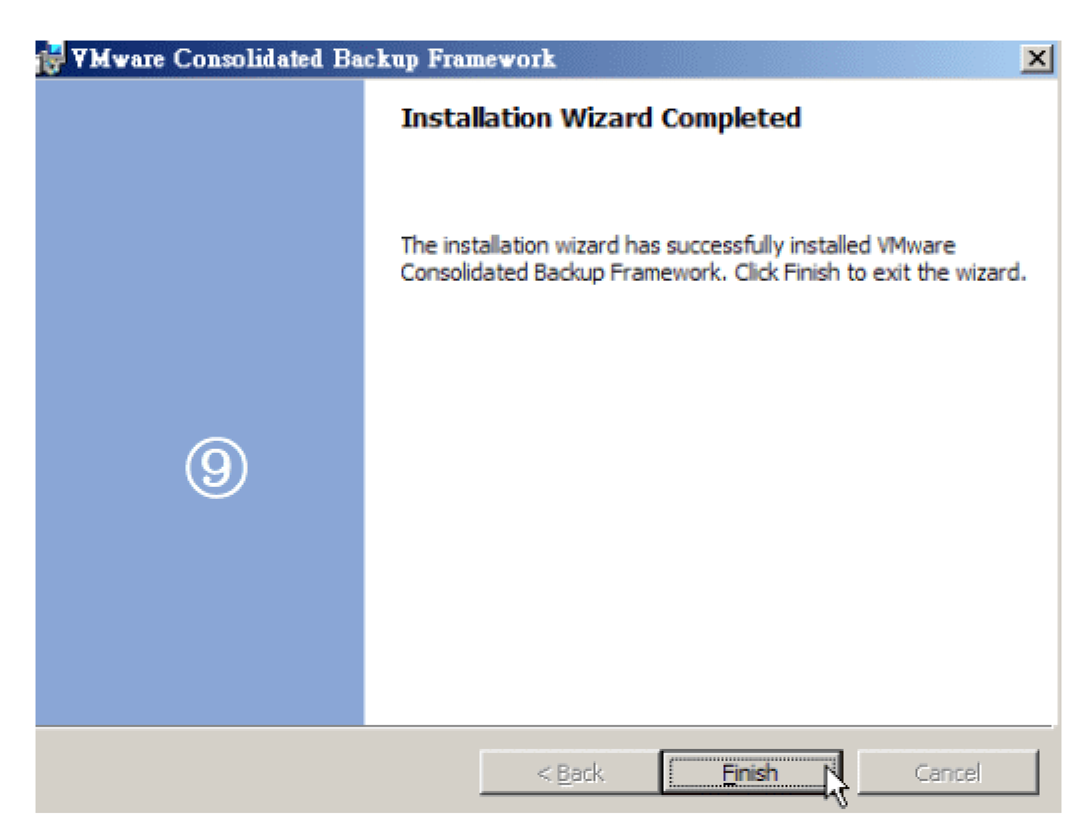
▲ 将文件克隆到硬盘



▲ 在安装时，会安装 VMware 的存储驱动程序

9. 当弹出画面时，安装落实，单击 Finish 按钮结束。

10. 我们可以在安装目录下看到 VCB 的文件。



▲ 安装落实

```
2010/06/16  下午 05:57    <DIR>          .
2010/06/16  下午 05:57    <DIR>          ..
2010/06/16  下午 05:57    <DIR>          config
2010/06/16  下午 05:57    <DIR>          generic
2009/02/26  下午 06:34           858,672 libeay32.dll
2009/02/26  下午 06:35         1,518,128 mountvm.exe
2003/03/18  下午 08:14           499,712 msvcp71.dll
2003/02/21  上午 04:42           348,160 msvcr71.dll
2009/02/26  下午 06:34           166,448 ssleay32.dll
2009/02/26  下午 06:35        10,656,304 types.dll
2009/02/26  下午 06:35               339 vcbCleanup.bat
2009/02/26  下午 06:35         1,391,152 vcbExport.exe
2009/02/26  下午 06:35         1,714,736 vcbMounter.exe
2009/02/26  下午 06:35           182,832 vcbSanDbg.exe
2009/02/26  下午 06:35         1,227,312 vcbSnapshot.exe
2009/02/26  下午 06:35               361 vcbSupport.bat
2009/02/26  下午 06:35         1,235,504 vcbVmName.exe
2009/02/26  下午 06:35         2,066,992 vmacore.dll
2009/02/26  下午 06:35         1,391,152 vmomi.dll
```

⑩

▲ 这就是 VCB 的文件及命令

2. 使用 VCB 备份示例

接下来我们要使用 VCB 的命令来备份，在备份之前，我们先定义以下的环境。

3. 使用 VCB 备份的环境

- vCenter Server 的 IP：192.168.1.243
- 要备份 VM 的 IP：192.168.1.168
- 登录 vCenter Server 的账号：administrator
- 登录 vCenter Server 的口令：PassWord
- 备份的种类：完整的 VM
- 备份目录：D:\Backup\Test

20.2.2 开始使用 VCB 命令

VCB 提供的命令很多，包括 vcbVMName.exe、vcbMounter.exe 等。最常用的命令包括下面几个：

1. 常用的 VCB 命令

- mountvm.exe：将 VM 的 VMDK 挂成逻辑磁盘。
- vcbExporet.exe：将 VCB 备份的 VMDK 导出的命令。
- vcbMounter.exe：真正运行备份的程序，以下会有说明。
- vcbSanDbg.exe：给出 SAN 及 iSCSI 的命令。
- vsbSnapshot.exe：制作快照的命令。
- vcbVmName.exe：取出 VM 参数的命令，以下会有介绍。

```
C:\Program Files\VMware\VMware Consolidated Backup Framework

2009/02/26  下午 06:35         1,518,128 mountvm.exe
2009/02/26  下午 06:35         1,391,152 vcbExport.exe
2009/02/26  下午 06:35         1,714,736 vcbMounter.exe
2009/02/26  下午 06:35           182,832 vcbSanDbg.exe
2009/02/26  下午 06:35         1,227,312 vcbSnapshot.exe
2009/02/26  下午 06:35         1,235,504 vcbVmName.exe
```

▲ 这是 VCB 提供的命令

2. 使用 VCB 命令：vcbVMName.exe 导出备份参数

安装好 VCB 之后，我们就可以使用 VCB 提供的命令来备份 VM。要执行 VCB 的命令，必须在 Windows 2003 下打开一个命令提示字符，并且进入 VCB 的目录，才能运行 VCB 提供的命令。

VCB 的备份命令很简单，主要有两个，一个是以 vcbVmName.exe 来确定 VCB 的备份参数，另一个则是 vcbVmName.exe 真正运行备份。我们就分别看看，首先是 vcbVmName.exe 的语法。

3. vcbVmName.exe 的语法

```
vcbVmName.exe -h <vCenter Server 的 IP 位置> -u <登录 vCenter Server 的用户账号> -p <登录 vCenter Server 的口令> -s ipaddr:<要备份 VM 的 IP 位置>
```

举例来说，我们要找出在 192.168.1.243 这个 vCenter Server 上有一个 VM，其内部 IP 为 192.168.1.168，那么使用 vcbVmName 找出其备份参数的命令。

4. 利用 vcbVmName 导出备份参数

```
vcbVmName.exe – h 192.168.1.243 – u administrator – p PassWord – s ipaddr:192.168.1.168
```

在运行完毕后，系统会提供一些参数，分别为 moref、name、uuid、ipaddr 等。这些参数可以选择其一来开始备份 VM，选定之后，我们就可以使用 vcbVmMount.exe 来备份 VM 了。

```
[2010-06-20 14:56:35.859 'BaseLibs' 2600 info] HOSTINFO: This machine has 1 phys
ical CPUS, 2 total cores, and 2 logical CPUs.
[2010-06-20 14:56:35.953 'BaseLibs' 2600 info] Using system libcrypto, version 9
0709F
[2010-06-20 14:56:36.531 'BaseLibs' 2600 warning] SSLVerifyCertAgainstSystemStor
e: Subject mismatch: forclone vs 192.168.1.243
[2010-06-20 14:56:36.531 'BaseLibs' 2600 warning] SSLVerifyCertAgainstSystemStor
e: The remote host certificate has these problems:

* The host name used for the connection does not match the subject name on the h
ost certificate

* The host certificate chain is not complete.
[2010-06-20 14:56:36.531 'BaseLibs' 2600 warning] SSLVerifyIsEnabled: failed to
read registry value. Assuming verification is disabled. LastError = 0
[2010-06-20 14:56:36.531 'BaseLibs' 2600 warning] SSLVerifyCertAgainstSystemStor
e: Certificate verification is disabled, so connection will proceed despite the
error
Found VM:
moref:vm-380
name:2003
uuid:420c21dd-5842-abed-4893-1abdc8f9f6a7
ipaddr:192.168.1.168

C:\Program Files\VMware\VMware Consolidated Backup Framework>
```

▲ 运行 vcbVmName 的结果，要注意图中注记的部分

5. 使用 VCB 命令：vcbMounter.exe 进行备份

VCB 提供的第二个命令 vcbMounter.exe 是真正提供备份的命令，在运行之后，要备份的 VM 会被制作一个快照挂上，并且开始进行备份。使用 vcbMounter.exe 的语法如下：

6. 使用 vcbMounter 的语法

```
vcbMounter.exe -h <vCenter Server 的 IP 位置> -u <登录 vCenter Server 的用户账号> -p <登录 vCenter Server 的口令> -s ipaddr:<要备份 VM 的 IP 位置> -t <备份种类 fullvm 或
```

```
是 file>  -r <备份的目的目录> -m <是否以网络为基础备份>
```

举例来说，我们要备份一个 IP 为 192.168.1.168 的 VM，那么要运行的命令就如下：

7. **备份 VM 的 vcbMounter 命令**

```
    vcbMounter.exe  -h  192.168.1.243  -u  administrator  -p  PassWord  -a
ipaddr:192.168.1.168 -t fullvm -r D:\Backup\Test -m nbd
```

- -h 192.168.1.243：vCenter Server 的 IP。
- -u administrator：登录该 vCenter Server 的账号。
- -p PassWord：登录该 vCenter Server 的口令。
- -a ipaddr:192.168.1.168：要备份 VM 的 IP。
- -t fullvm：备份整个 VM。
- -r D:\Backup\Test：将备份的文件放在 D:\Backup\Test 这个目录中。
- -m nbd：使用以网络为基础的方式备份，而不直接可写 SAN 中。

当开始备份时，系统会弹出备份的进度，其中最大的文件一定是 VMDK，因此会花较长的时间。一般来说，使用 vcbMounter 备份，通常会花上数十分钟到几小时的时间。

```
Copying "[LUN1] 2003//vmware-31.log":
        0%=====================50%=====================100%
        **************************************************

Copying "[LUN1] 2003//vmware-32.log":
        0%=====================50%=====================100%
        **************************************************

Copying "[LUN1] 2003//vmware-28.log":
        0%=====================50%=====================100%
        **************************************************

Copying "[LUN1] 2003//vmware-29.log":
        0%=====================50%=====================100%
        **************************************************

Copying "[LUN1] 2003//vmware-30.log":
        0%=====================50%=====================100%
        **************************************************

Copying "[LUN1] 2003//vmware.log":
        0%=====================50%=====================100%
        **************************************************
```

▲ 这是使用 vcbMounter 的运行结果，其中 VMDK 会最久

20.2.3 使用 VCB 还原 VM

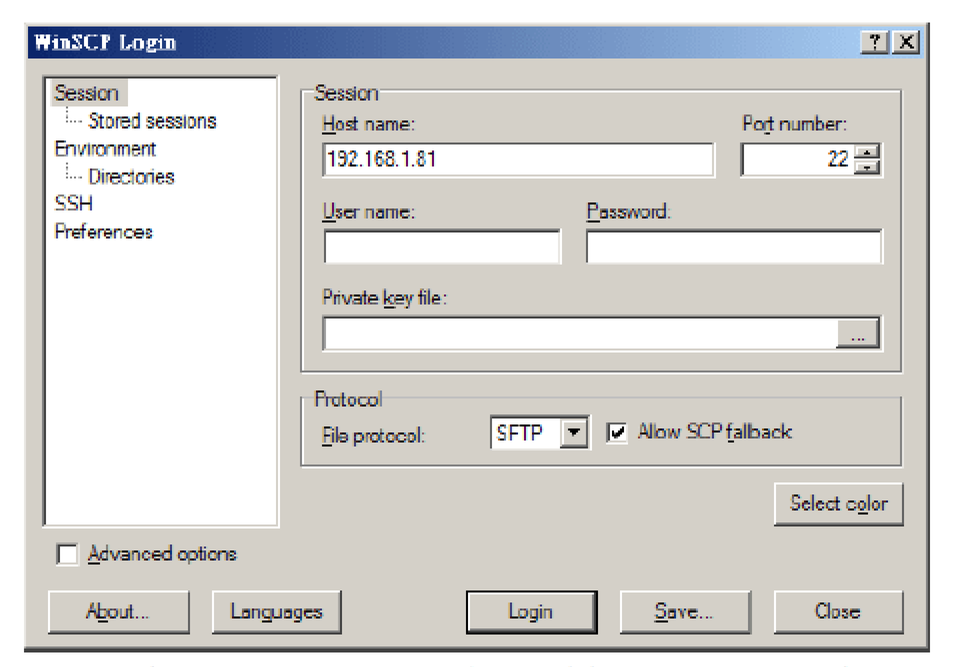
备份之后当然就是还原了。使用 VCB 还原的手续较单纯，只要先将整个备份的目录传到 ESX/ESXi 的主机上，再使用 VCB 在 ESX/ESXi 主机上的命令 vcbRestore 来还原。

1. **将 VM 的备份文件上传至 ESX 主机**

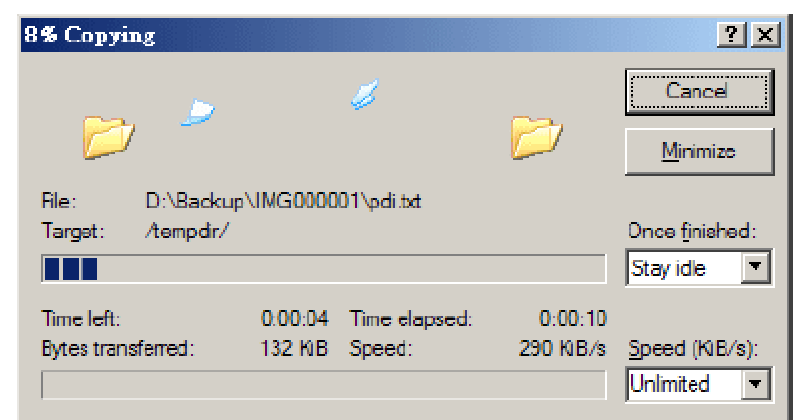
要将文件上传到 ESX，我们可以使用 WinSCP 这个软件。读者们可以查阅第 9 章有关 WinSCP 的安装及使用。

▶ **将 VCB 备份的文件上传至 ESX**

1. 进入 WinSCP，并且将使用 VCB 备份的整个目录上传至 ESX。
2. 此时会开始传输，由于文件较大，因此会较花时间。

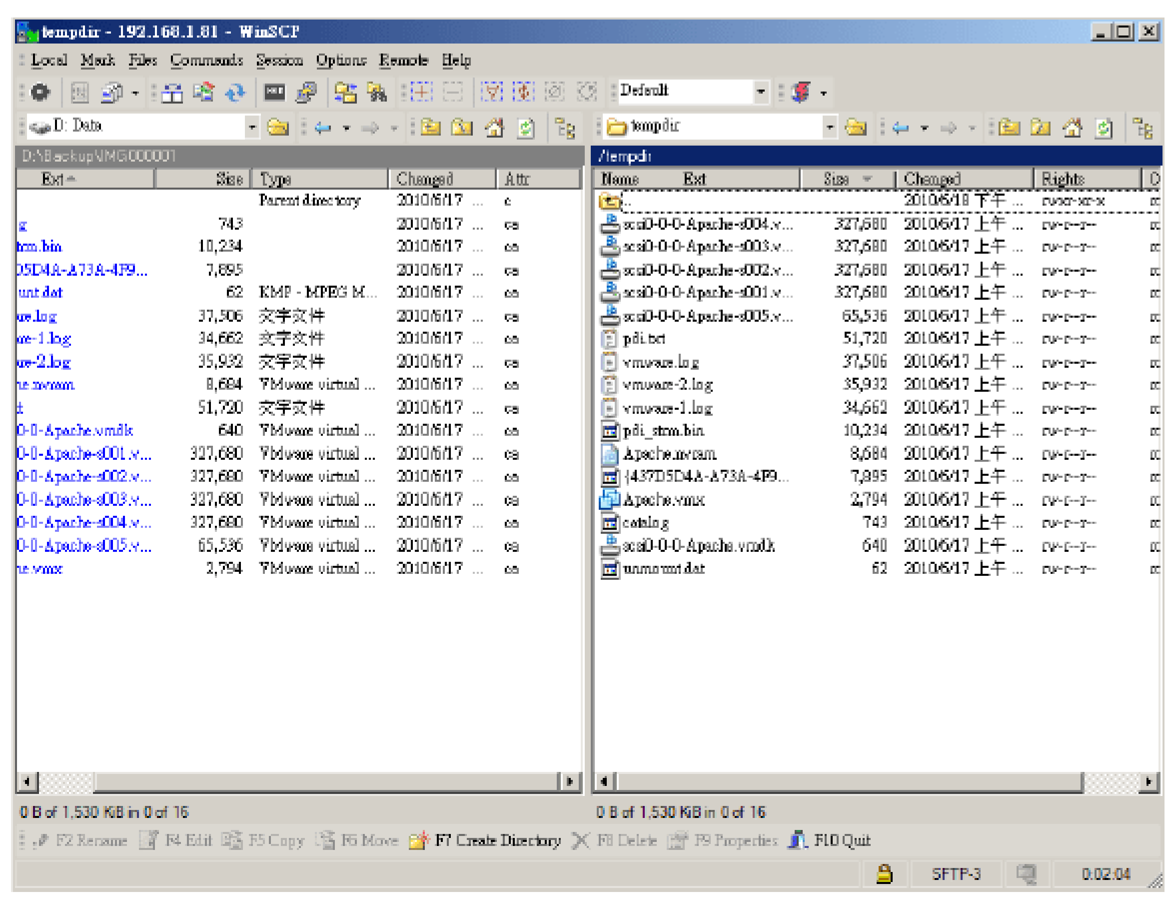


▲ 进入 WinSCP，并且联机到 ESX 主机上



▲ 开始传输

3. 传输完毕之后，我们可以在 ESX 的 Service Console 下看到这些文件。



▲ 传输到 ESX 主机上了

2. 使用 VCB 的命令还原 VM

当我们将文件传到 ESX 主机之后就可以进行还原了。还原 VCB 备份的文件，在 ESX 下也是使用标准的命令。

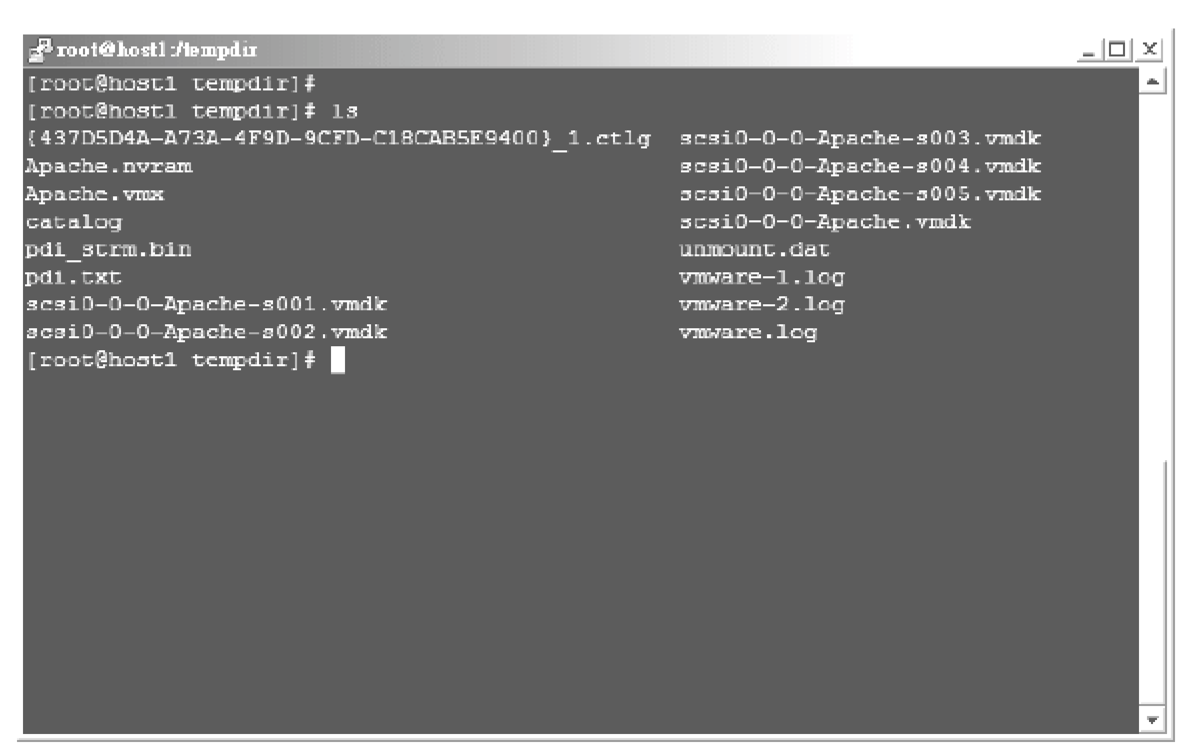
▶ 使用 vcbRestore 来还原 VM

1. 首先我们使用 PuTTY 进入 ESX 主机，PuTTY 的使用也请引用第 9 章的说明。

```
login as: root
root@192.168.1.81's password:
Last login: Fri Jun 18 18:06:08 2010 from 192.168.1.200
[root@host1 ~]#
[root@host1 ~]#
```

▲ 使用 PuTTY 进入 ESX 主机

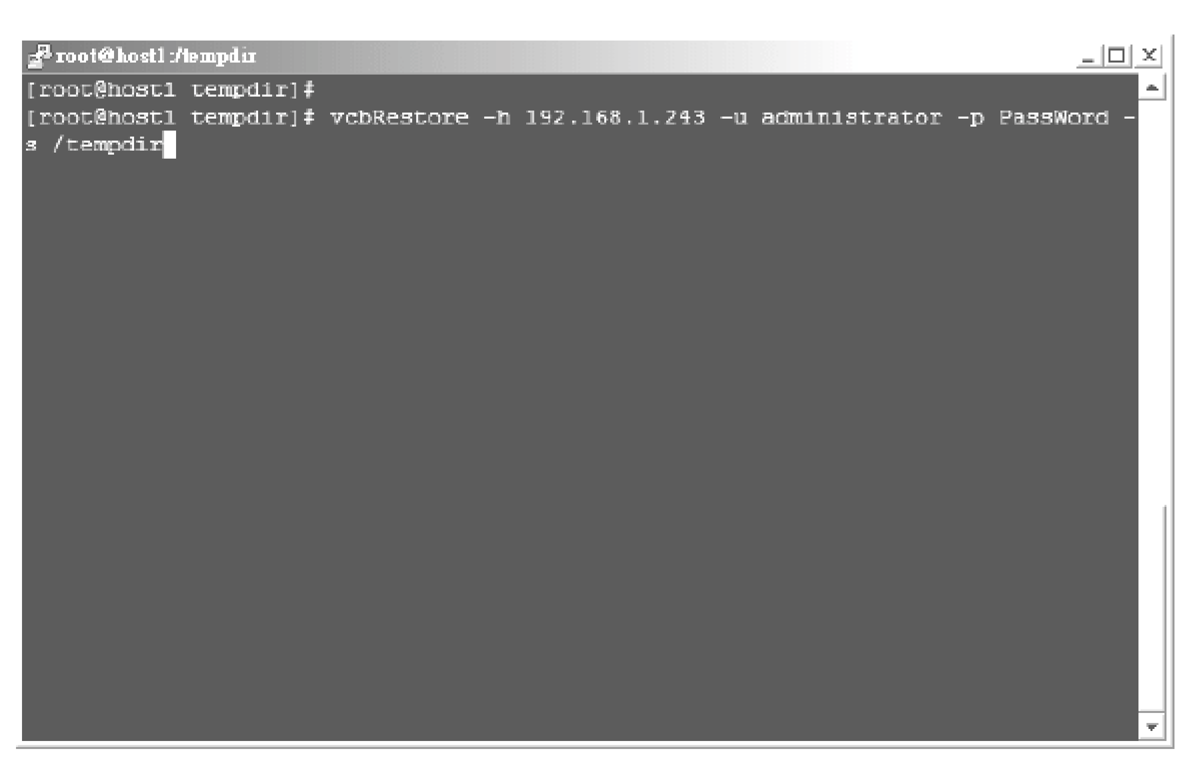
2. 进入之后，首先找到刚才上传的整个目录，如/tempdir 这个目录。



▲ 进入刚才上传的目录

3. 接下来键入还原命令“vcbRestore－h <vCenter Server 的 IP> -u <用户账号> -p <用户口令> -s <暂存目录>”。

4. 我们键入 vcbRestore －h 192.168.1.243 －u administrator －p PassWord －s /tempdir。



▲ 运行还原的动作

5. 此时会开始进行还原，直至还原成功为止。当还原成功之后，可以在 Datastore 看到这个 VM。

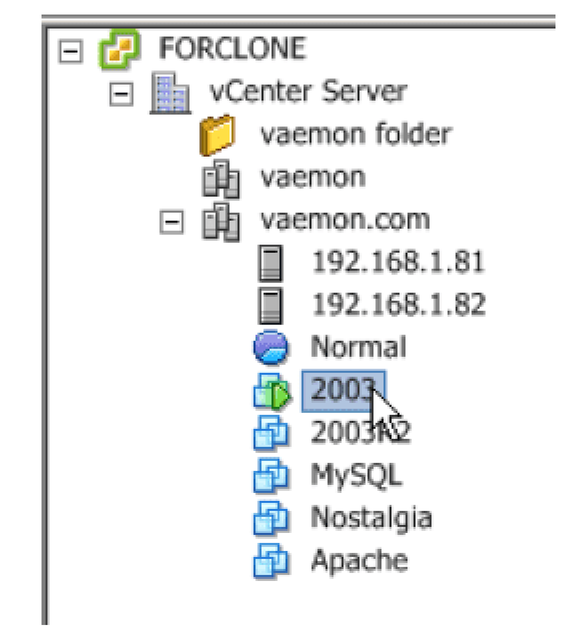
```
root@host1:/tempdir
r
[2010-06-18 18:09:09.673 F4F7EB60 error 'vcbRestore'] Error: Invalid user name o
r password.
[root@host1 tempdir]# vcbRestore -h 192.168.1.243 -u administrator -p
g -s /tempdir
[2010-06-18 18:09:20.240 F4F39B60 info 'App'] Current working directory: /tempdi
r


VMware ESX Question:
Importing the virtual disk
  "scsi0-0-0-Apache.vmdk"
would overwrite the existing file
  "[LUN0] Apache/Apache.vmdk".
How do you want to proceed?
0) Abort Restore
1) Keep Existing
2) Replace Existing

Please choose a number [0-2]: 2

Converting "/vmfs/volumes/LUN0/Apache/Apache.vmdk" (VMFS (flat)):
        0%--------------------50%--------------------100%
        *****************
```

▲ 还原落实



▲ 在 vSphere Client 中就可以看到这个 VM 了

20.3 使用第三方软件备份还原

虽然 vSphere 只在入门版本建议使用 VCB，但各大备份软件厂家无不利用 VCB Framework 的架构推出专门备份 vSphere VM 的产品。以 Windows 系统而言，我们可以将第三方备份软件并发安装在 VCB Backup Proxy 上，这样就可以使用 VCB Framework 将 VM 的数据备份出来。在这一小节我们就来介绍最常见的 Symantec Backup Exec 2010 这个产品。

20.3.1 安装及配置 Symantec Backup Exec 2010

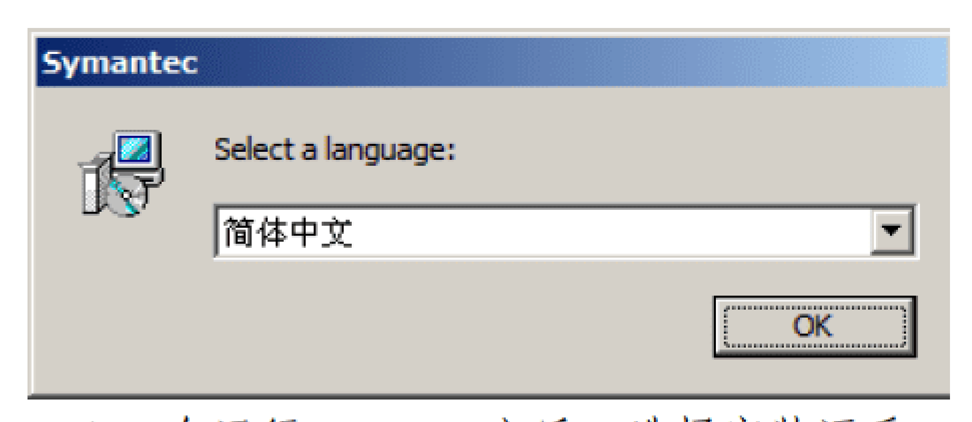
在使用 VCB 这个 Framework 之前，当然要先安装及配置 Symantec Backup Exec 2010。

1. 安装 Symantec Backup Exec 2010

首先我们要确定 VCB 还没安装，VMware 的建议是先安装备份软件再安装 VCB。因此我们先找一台 Windows 2003 来安装 Symantec Backup Exec 2010。

► 安装 Symantec Backup Exec 2010

1. 将 Symantec Backup Exec 2010 的光盘放入光驱，此时会激活自动运行，在第一个画面选择“简体中文”。



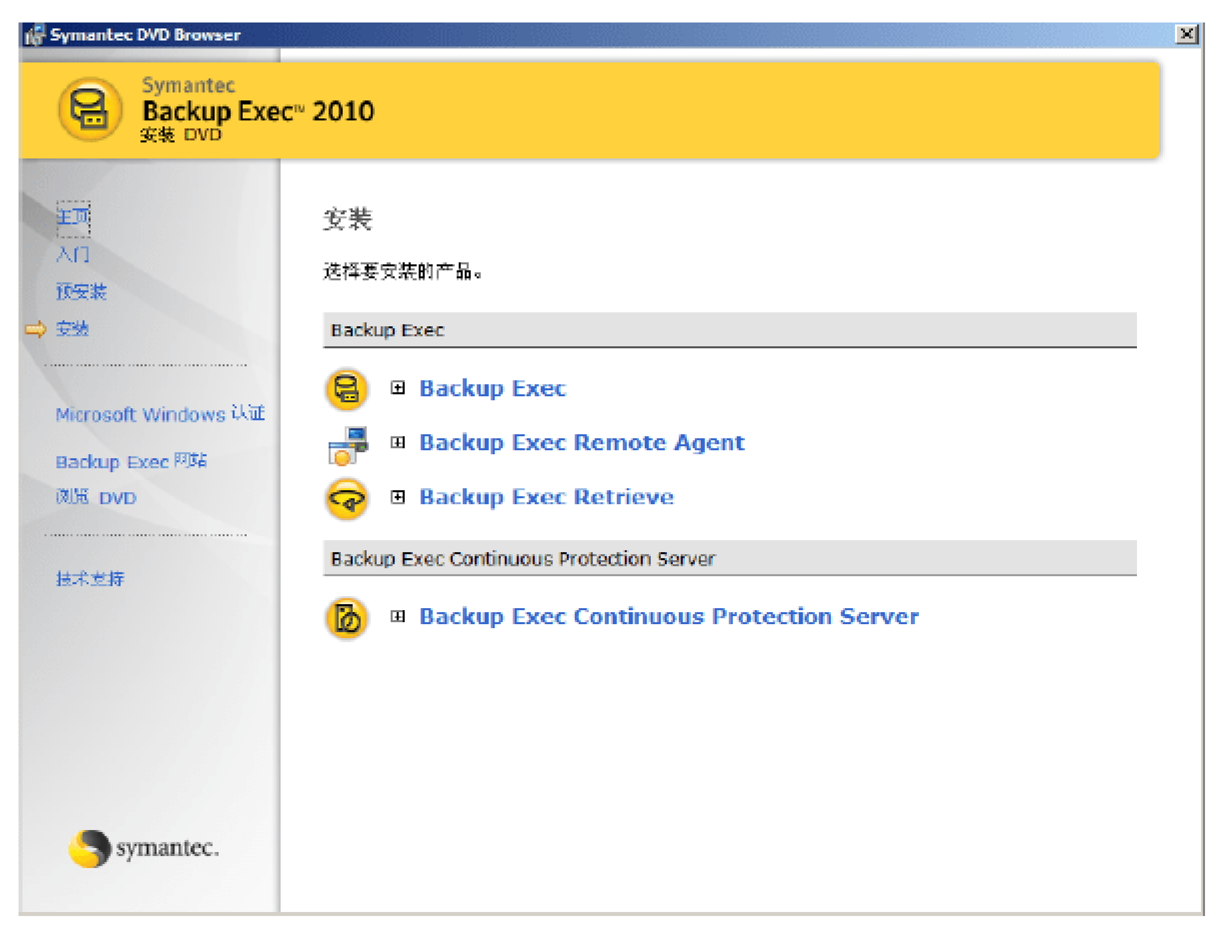
▲ 在运行 Autorun 之后，选择安装语系

2. 接下来我们就直接选择“安装产品”选项。



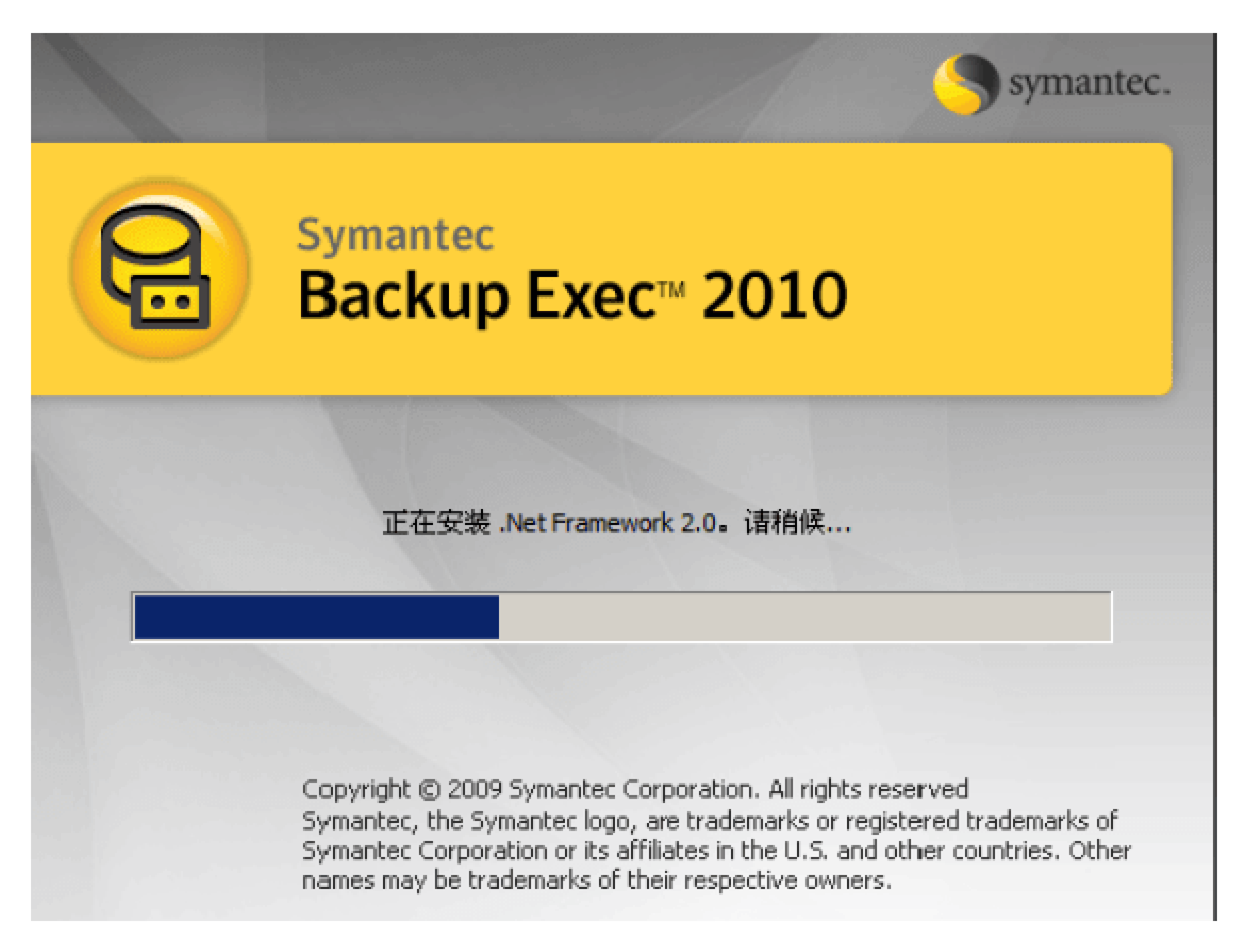
▲ 选择第三项安装产品

3. Symantec Backup Exec 2010 主要产品就是 Backup Exec，因此选择第一项。



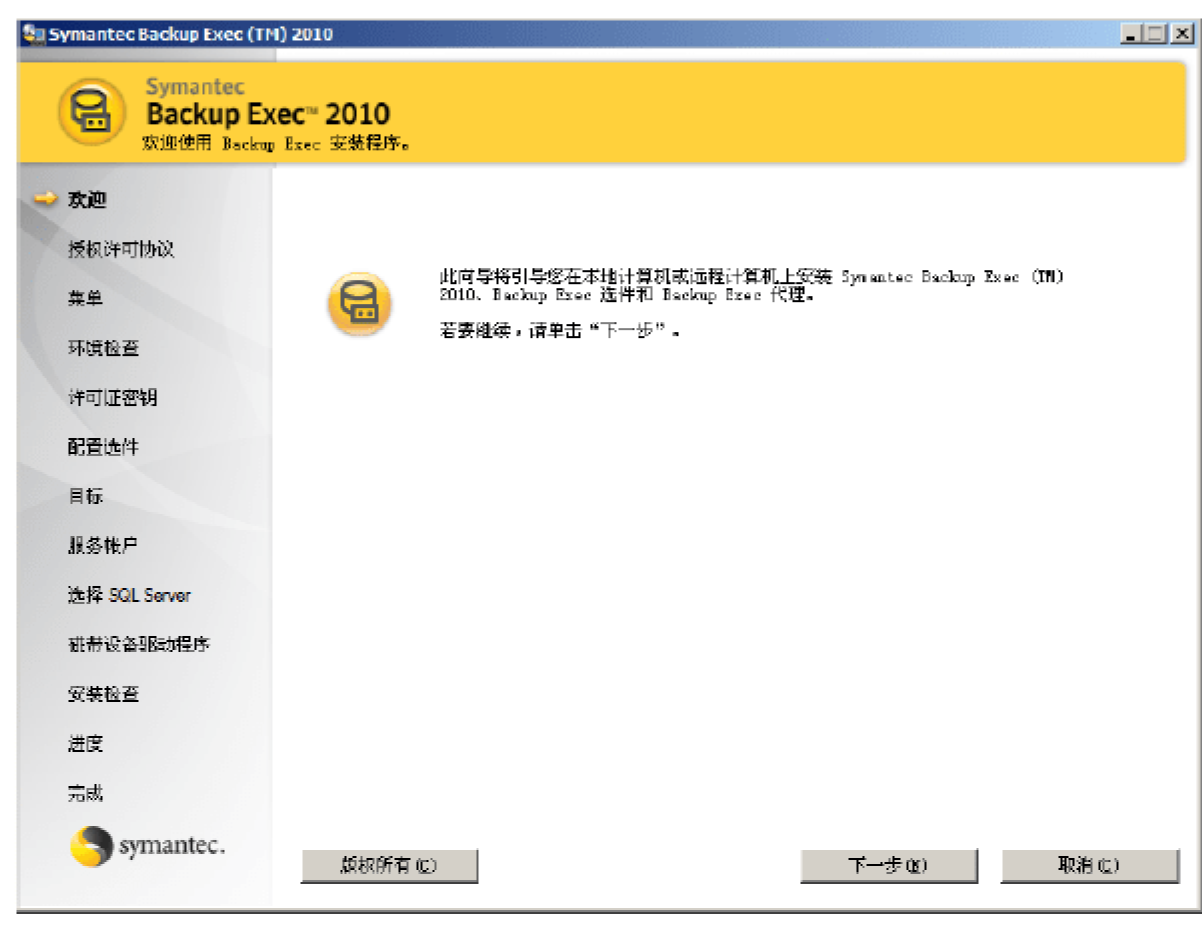
▲ 选择第一项的安装

4. 此时系统会装载安装文件。



▲ 开始装载安装文件

5. 接下来会进入安装向导，单击“下一步”按钮。



▲ 进入安装向导

6. 选择第一项同意授权，单击“下一步”按钮。



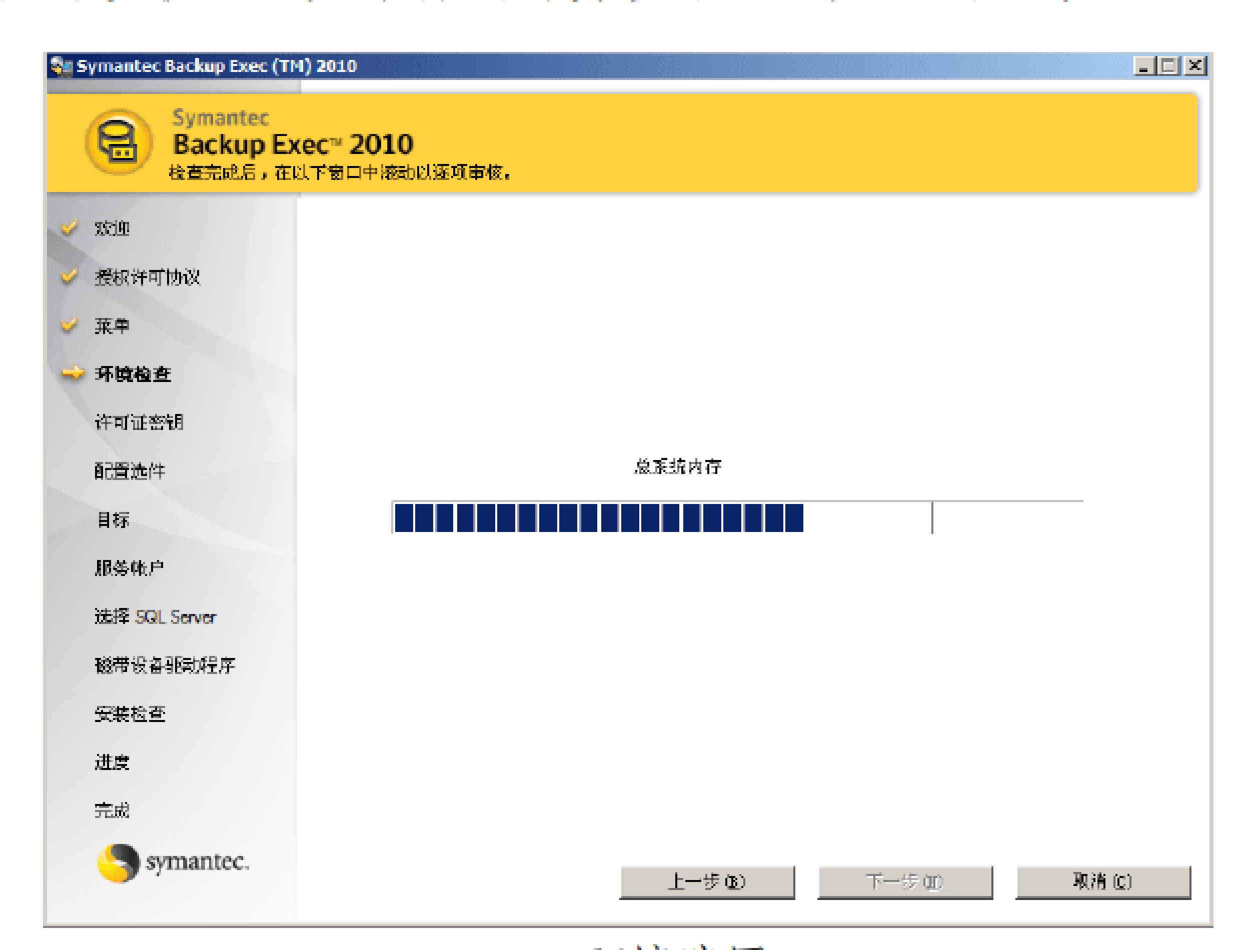
▲ 同意授权

7. 选择本地安装，并且选择第一项安装软件及选项。单击“下一步”按钮。



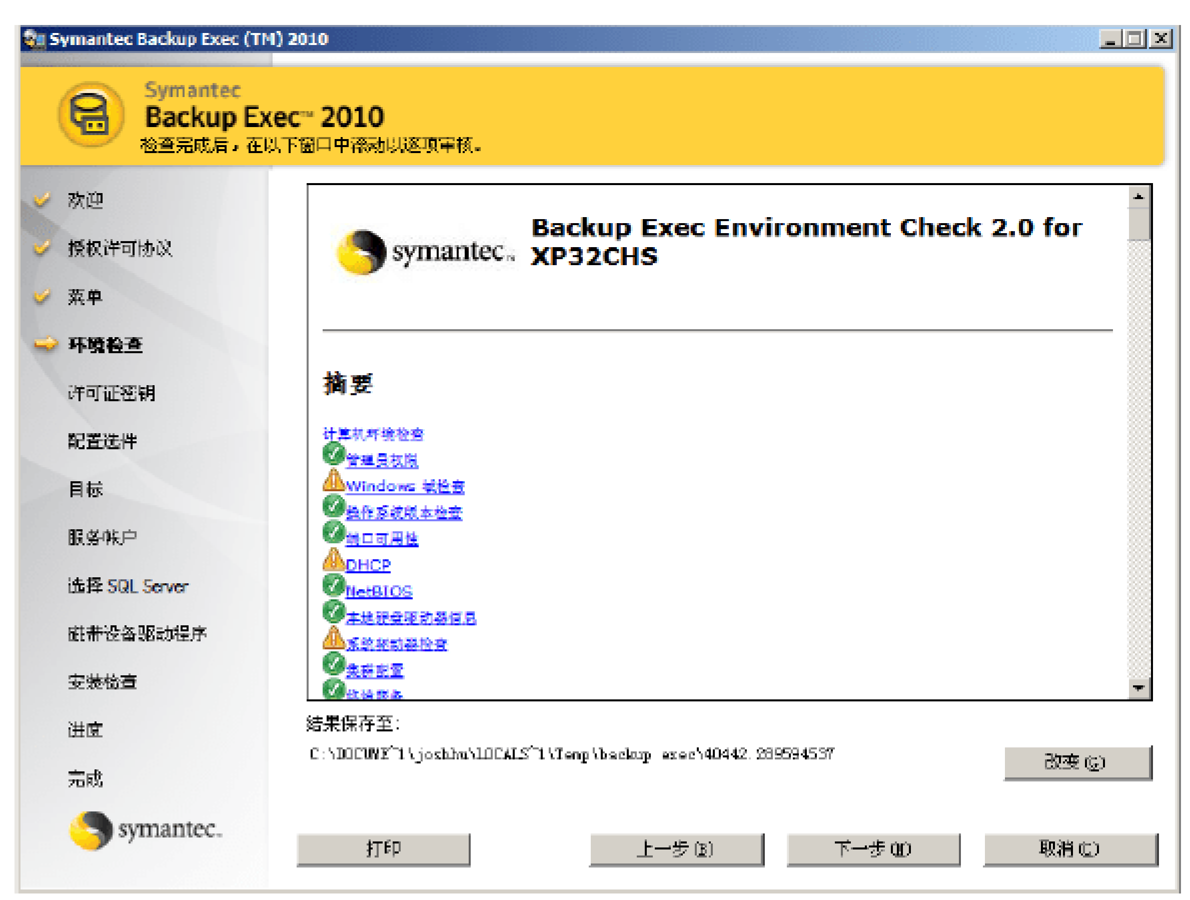
▲ 选择本地安装的第一项

8. 此时会遍历环境状态。如果你的计算机不匹配产品的基本组态，会弹出错误消息。



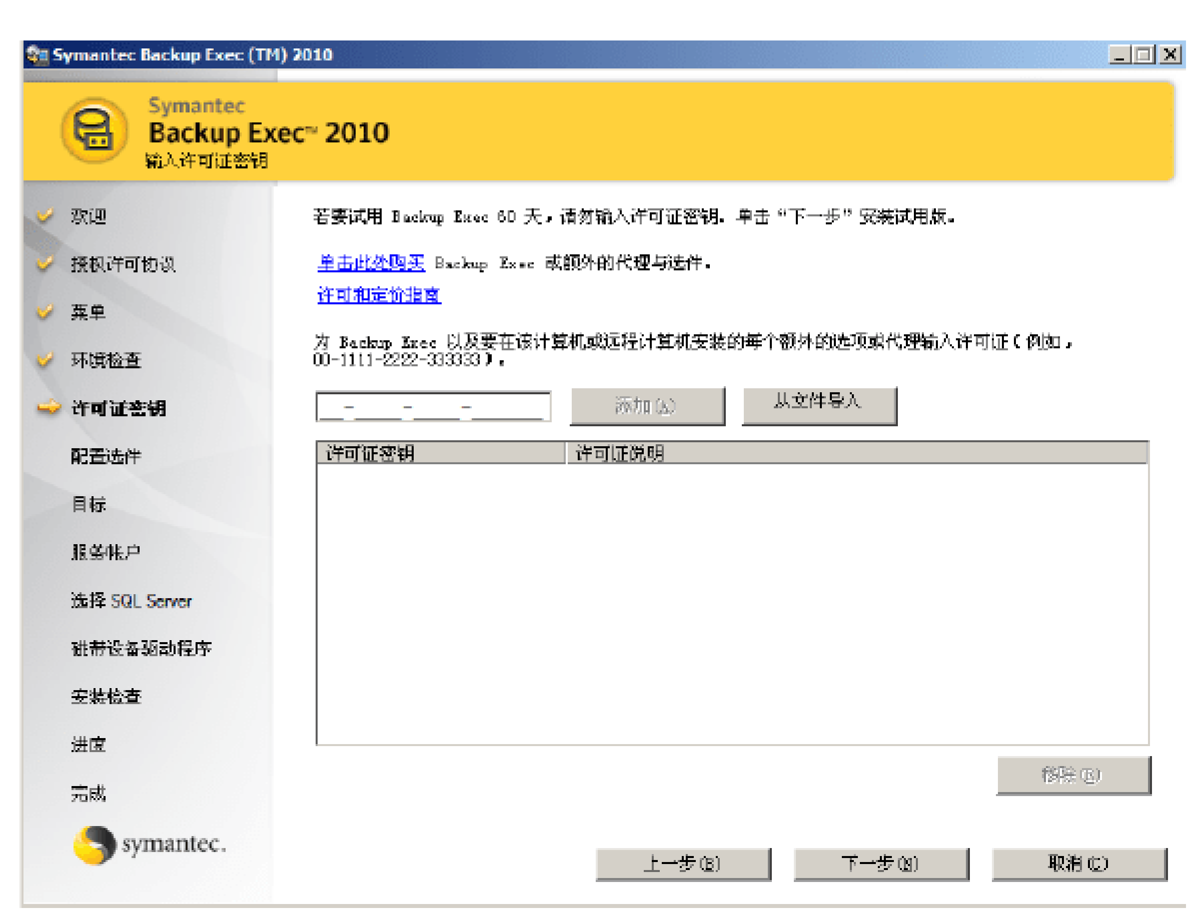
▲ 环境遍历

9. 此时会弹出遍历结果，大部分的 Windows 2003/2008 主机都能正确安装。单击“下一步”按钮。



▲ 弹出遍历结果，如果不匹配是无法进入下一步的

10. 接下来是键入序号，我们可以单击“下一步”按钮，以试用版的方式安装。



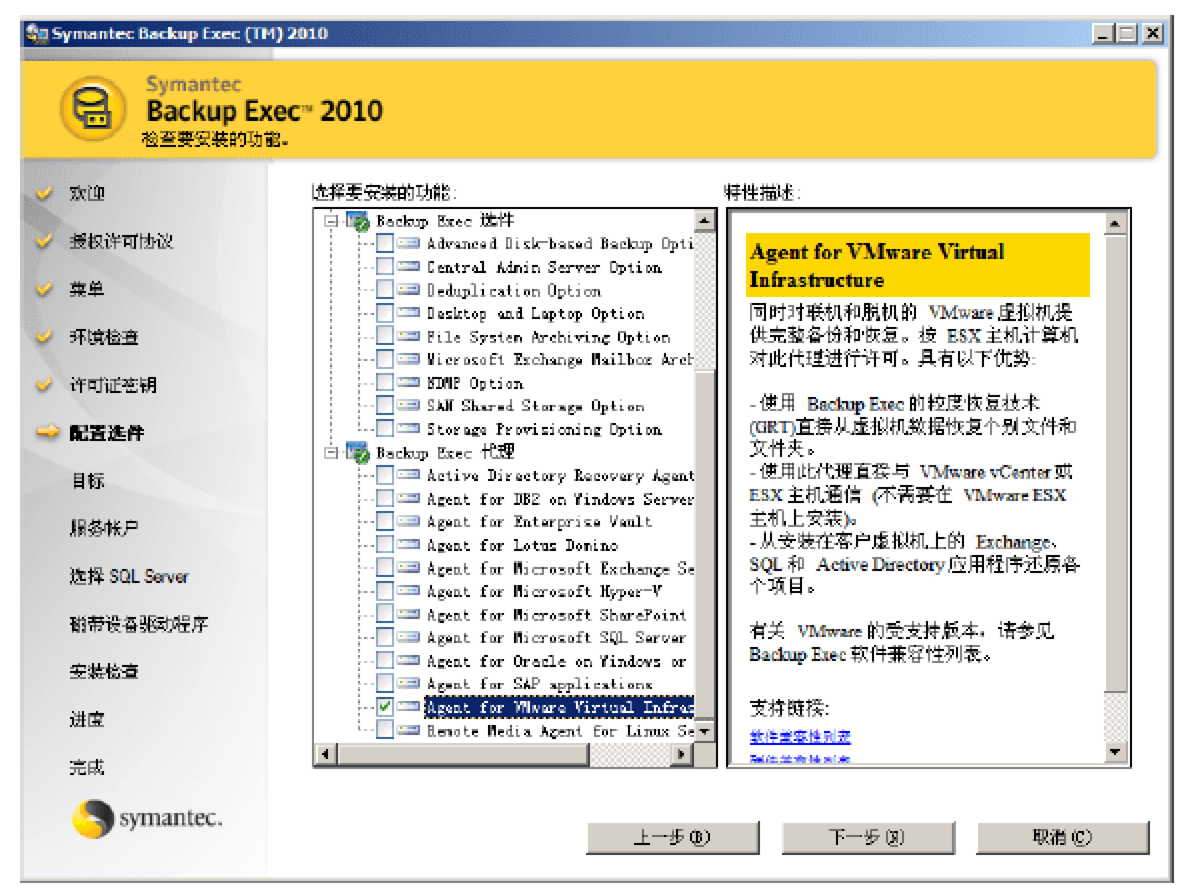
▲ 不键入序号可以使用 30 天，以试用版的方式运行

11. 试用版拥有大部分的功能，包括 VMware 的备份。



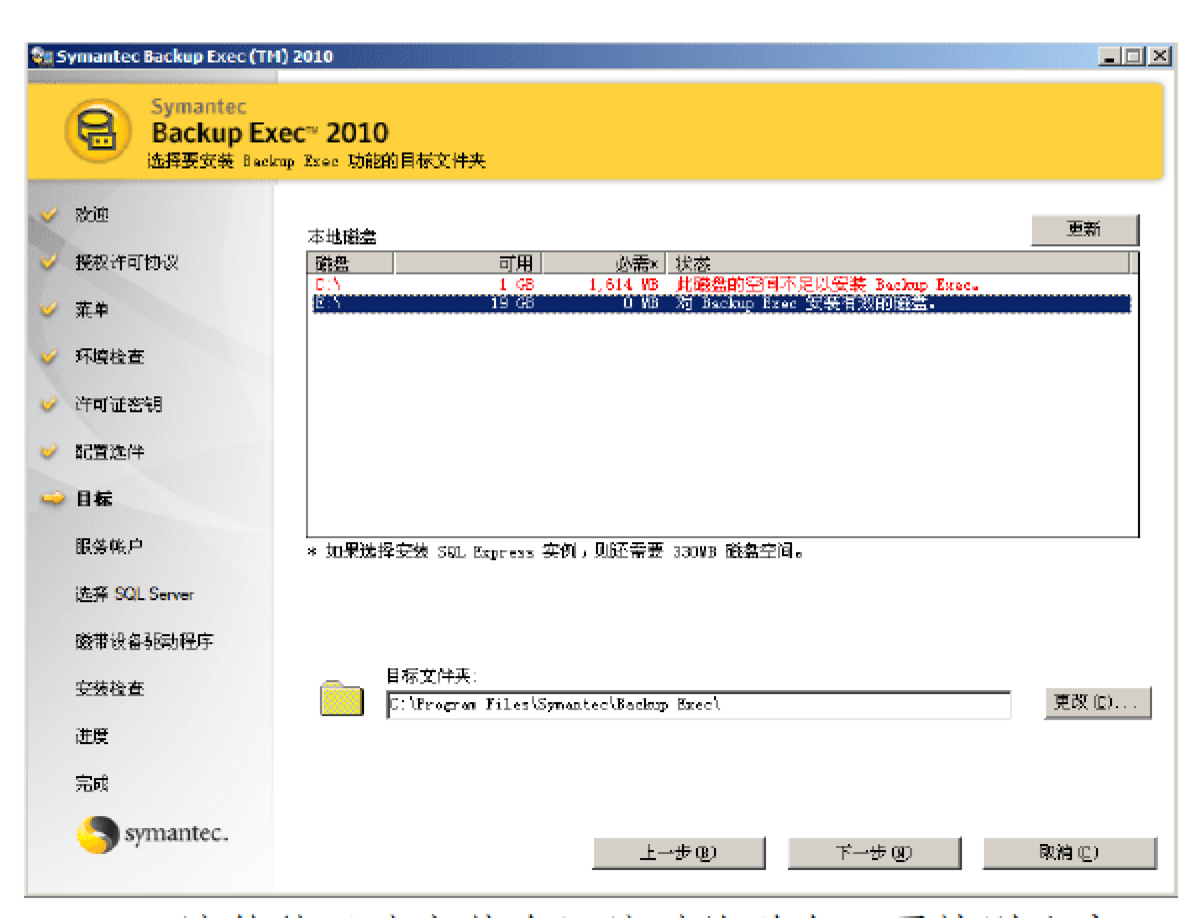
▲ 该软件有许多功能是需要单独购买的

12. 接下来是选择安装功能，千万别忘了在 Backup Exec Agent 的地方选择 Agent for VMware Virtual Infrastructure。单击“下一步”按钮。



▲ 千万要选择 VMware 这一项

13. 接下来是选择安装的目的。注意这个产品无法安装在压缩过的磁盘分区。单击“下一步”按钮。



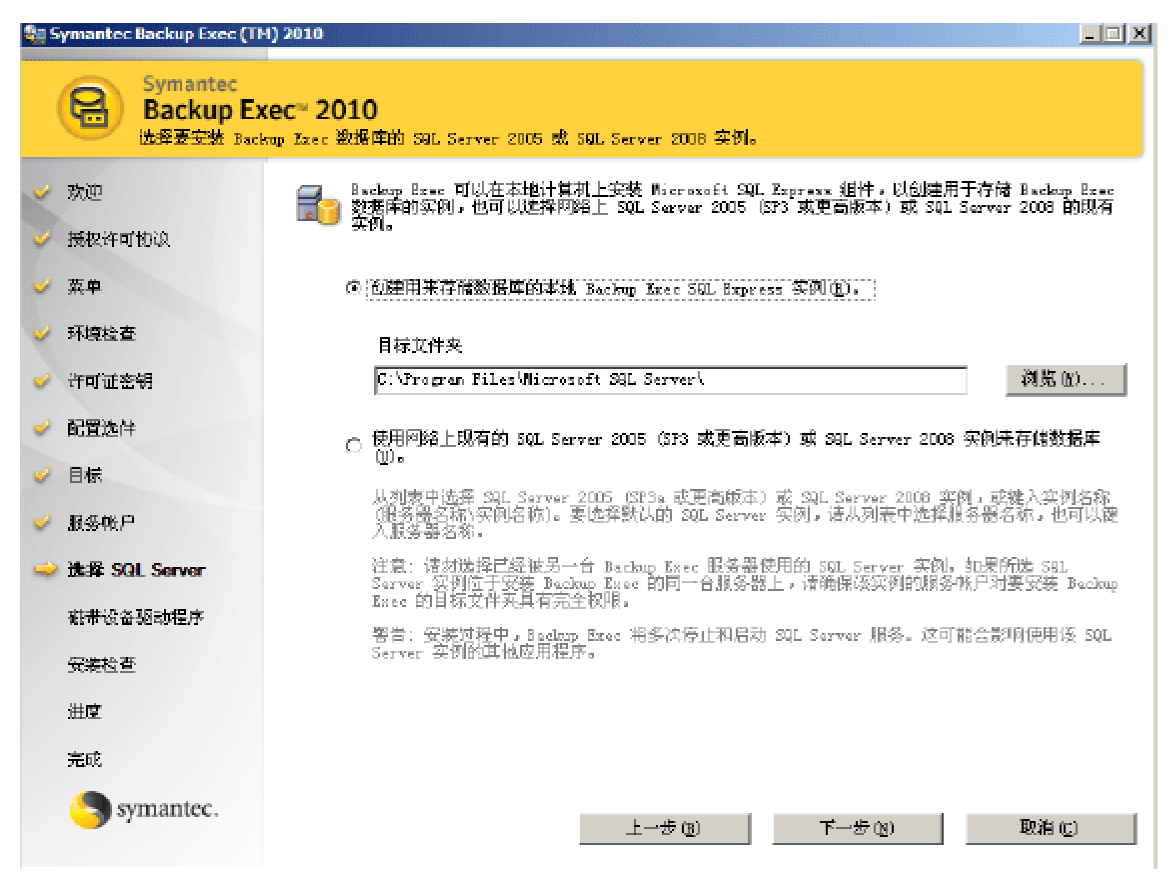
▲ 该软件无法安装在压缩过的磁盘，要特别注意

14. 接下来键入本地的用户账号口令。由于 Backup Exec 会在引导时装载许多服务，因此必须使用系统管理员的账号。单击“下一步”按钮。



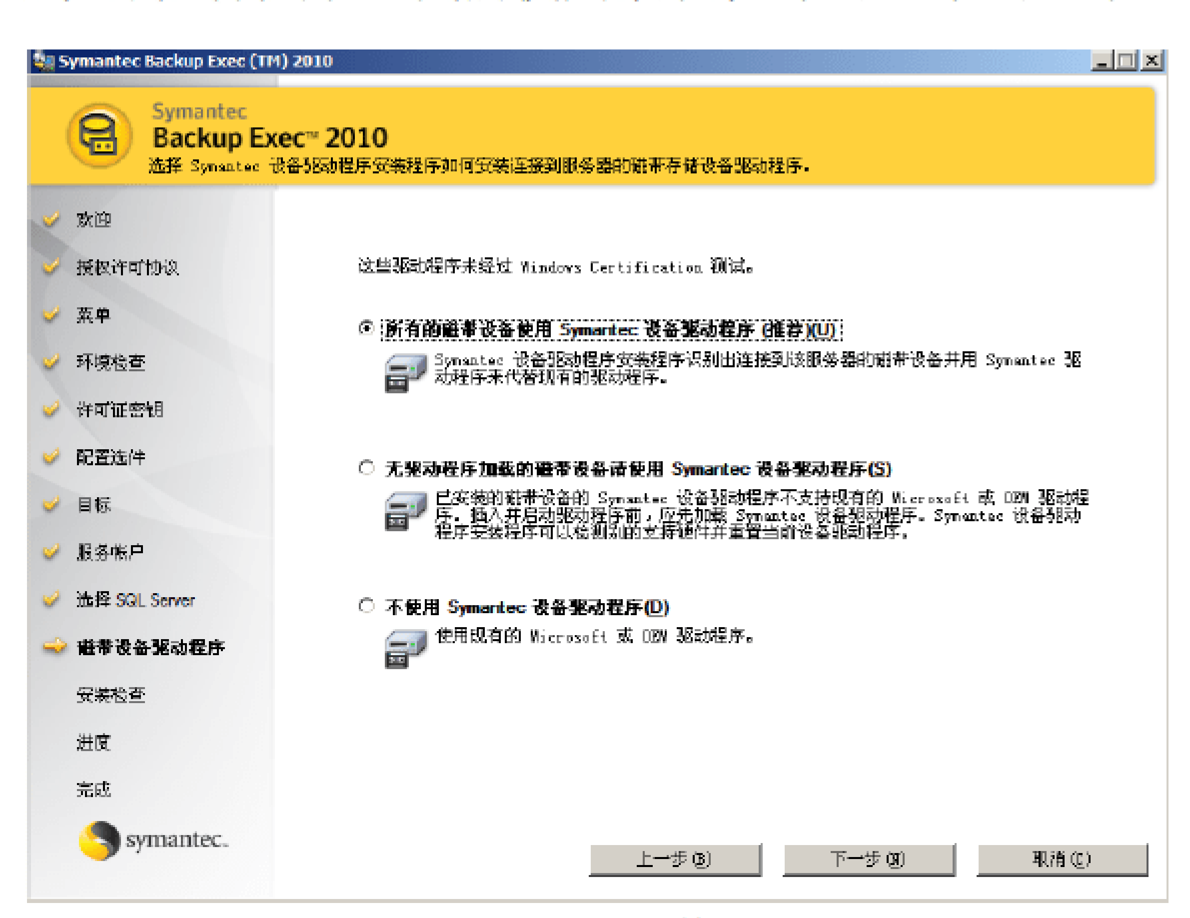
▲ 键入账号口令，是在 Windows 2003 下激活时需要，不是 VMware 的

15. 接下来是安装 SQL Server，如果你的计算机已经有 SQL 了，就可以使用原有的数据库。单击“下一步”按钮。



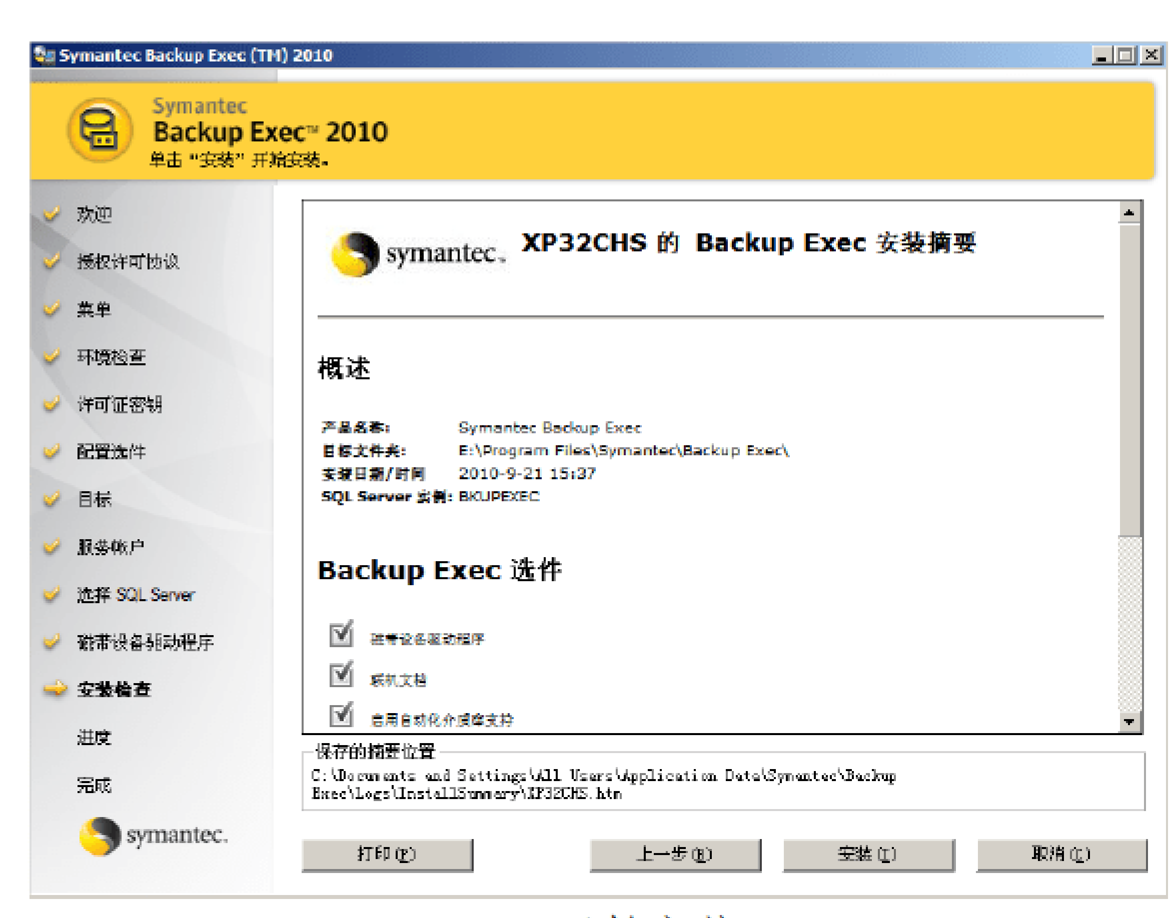
▲ 安装 SQL

16. 接下来是磁带的驱动程序，我们使用默认值即可。单击“下一步”按钮。



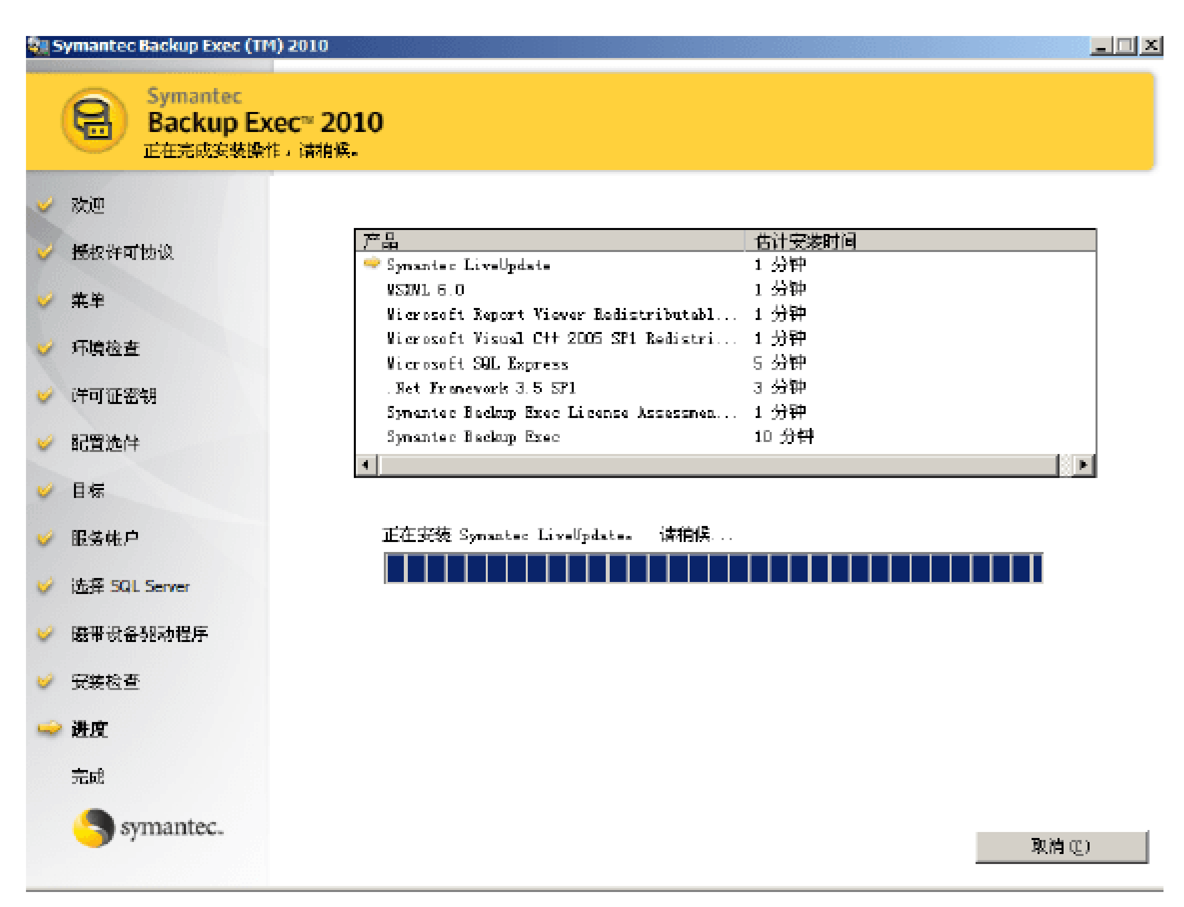
▲ 直接选择第一项

17. 当搜集完所有的信息之后，单击“安装”按钮即进行安装。



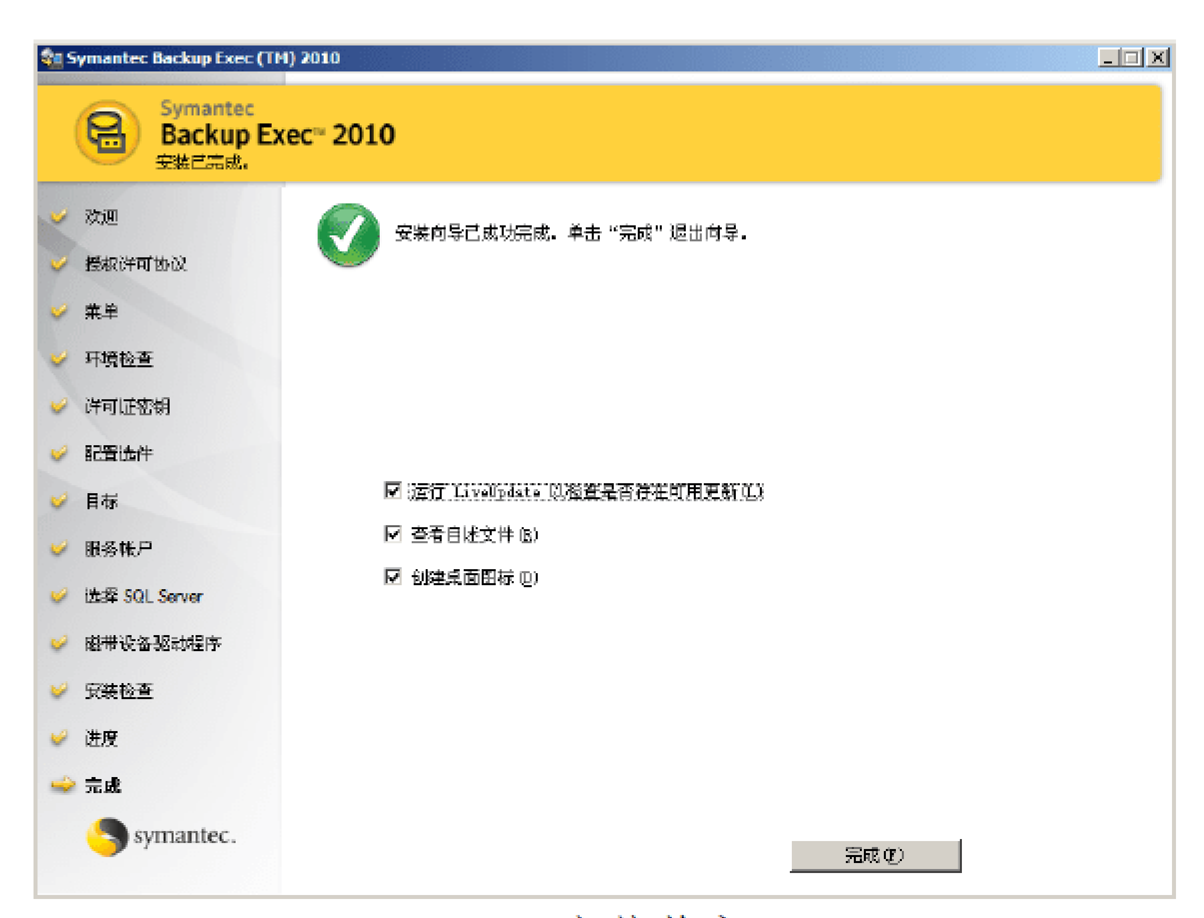
▲ 开始安装

18. 接下来是安装产品，大约需要 20 分钟可以安装落实。



▲ 大约需要 20 分钟可以安装落实

19. 当安装落实后，单击“完成”按钮即可。

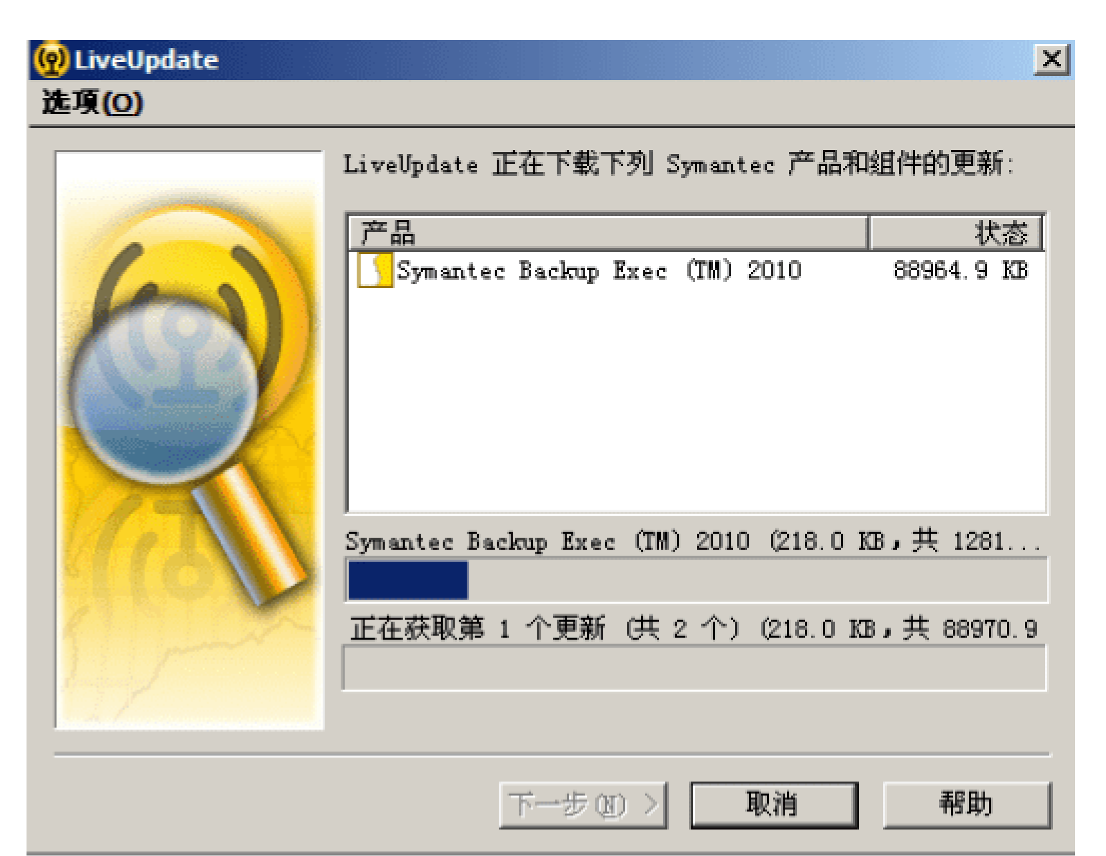


▲ 安装落实

20. Symantec 的产品通常会有 LiveUpdate，会替产品补丁，这里先将产品补丁到最新的版本。接下来是补丁所有的条目。单击“下一步”按钮。



▲ 别忘了补丁



▲ 开始补丁，补丁完毕后即可使用

接下来会开始补丁，补丁完即可使用了。

2. 创建管理 VM 的用户账户

要从 Symantec Backup Exec 2010 备份 VM，必须让 Symantec Backup Exec 2010 拥有访问 VM 的权利。由于 Symantec Backup Exec 2010 是通过 VCB 这个 Framework 来管理 vCenter Server，因此首先要在 Symantec Backup Exec 2010 中创建管理账户，我们就来看看具体步骤。

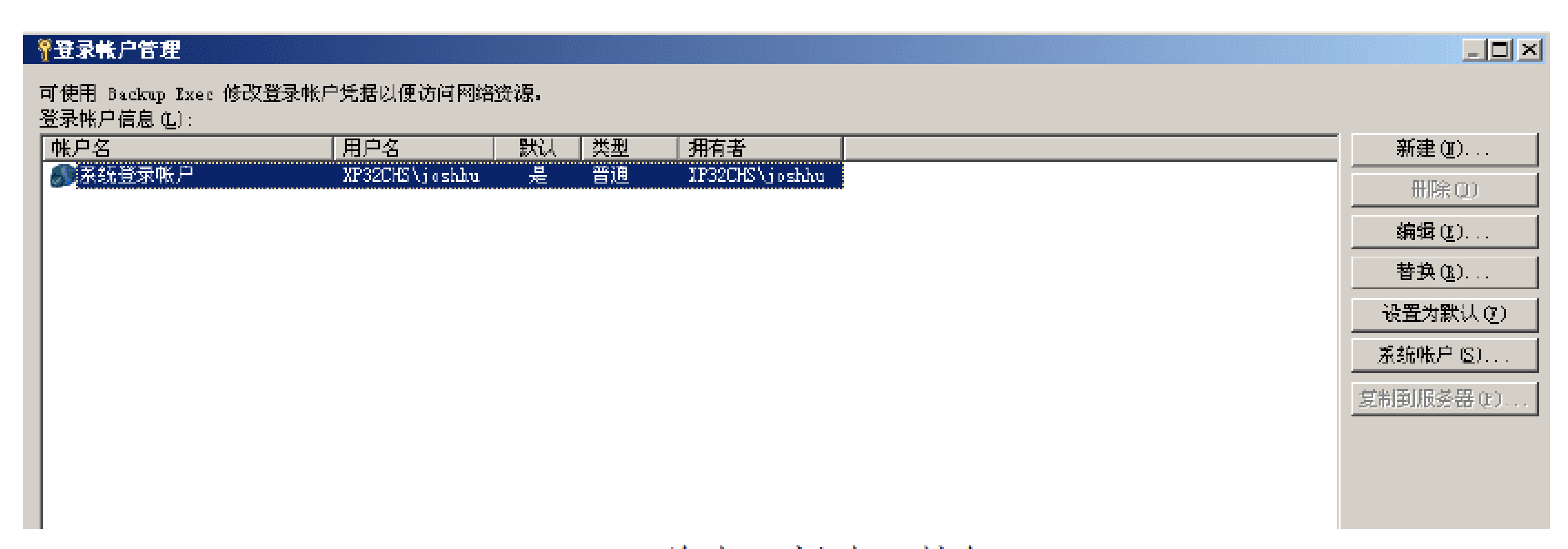
▶ 在 Symantec Backup Exec 2010 中创建管理账户

1. 进入 Symantec Backup Exec 2010 之后，选择“网络/登录账户”选项。



▲ 选择“登录账户”选项

2. 在登录账户的画面中单击“新建”按钮。



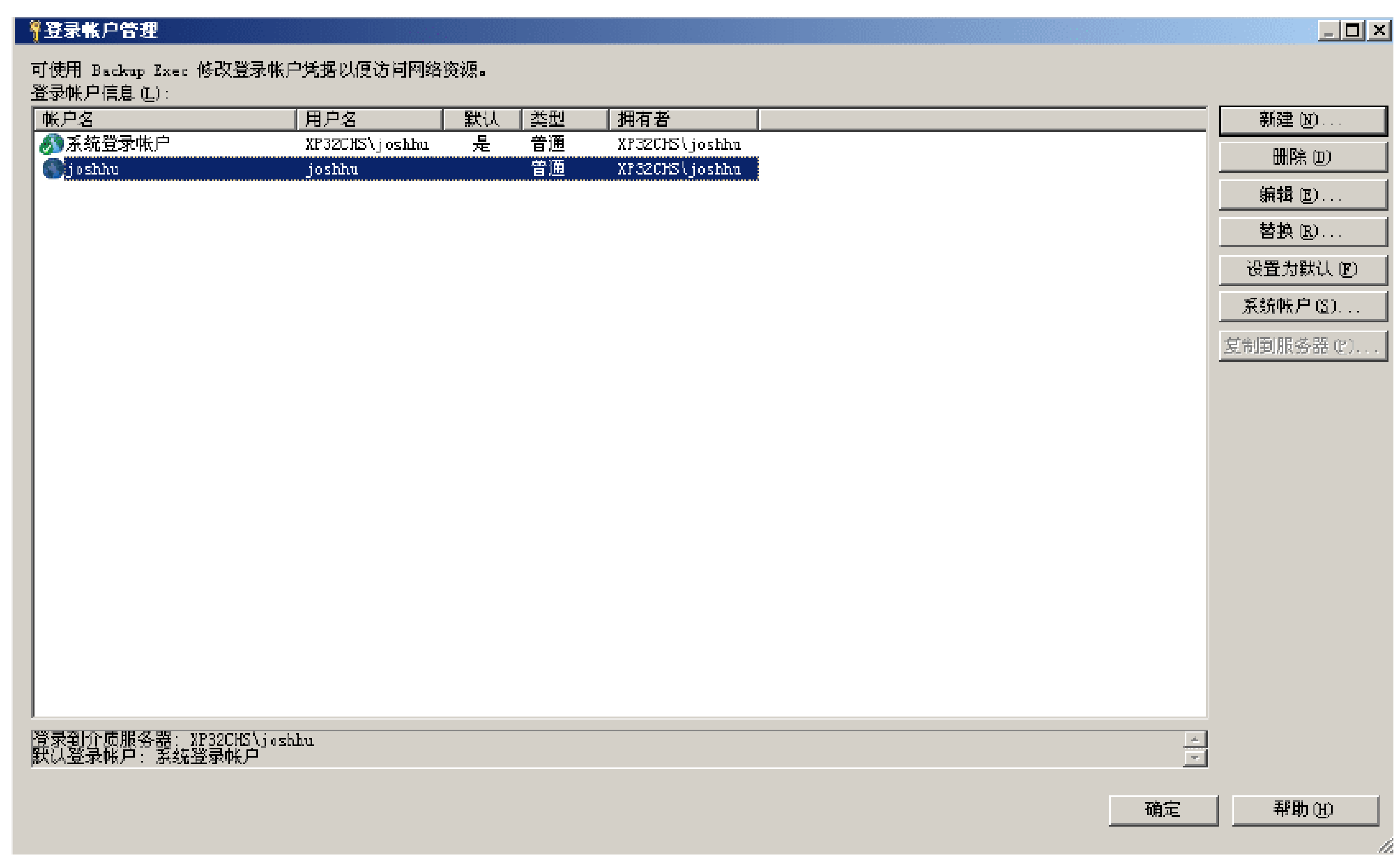
▲ 单击“新建”按钮

3. 此时我们要键入登录 vCenter Server 的账号及口令。键入完单击“确定”按钮。



▲ 这里键入的是 vCenter Server 的账号及口令

4. 这时可以在登录账户管理的画面中看到登录的账户了。单击“确定”按钮回到主画面。



▲ 新建完毕

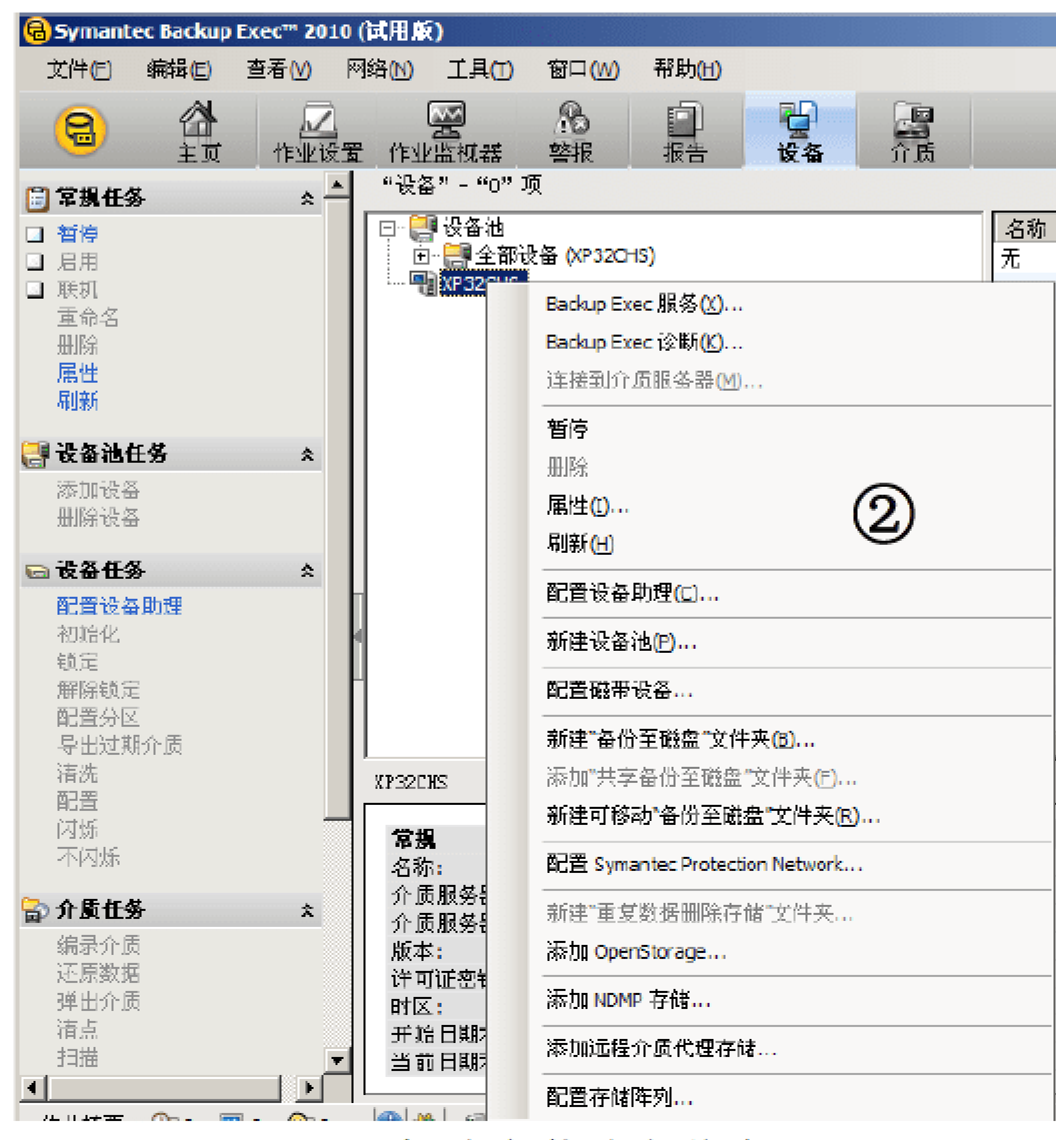
3. 在 Symantec Backup Exec 2010 下加入备份设备

Symantec Backup Exec 2010 是一个十分强大的备份软件，可以将数据备份到任何其能抓到的硬件之下。但在备份 VM 之前，我们必须先定义备份的目的地。一般来说，我们可以将 VM 备份到本地磁盘，也可以备份到网络磁盘上，我们在这里就试着加入 NFS 网络磁盘当作备份的目的。

▶ 在 Symantec Backup Exec 2010 加入备份设备

1. 进入 Symantec Backup Exec 2010，并选择“设备”选项卡。

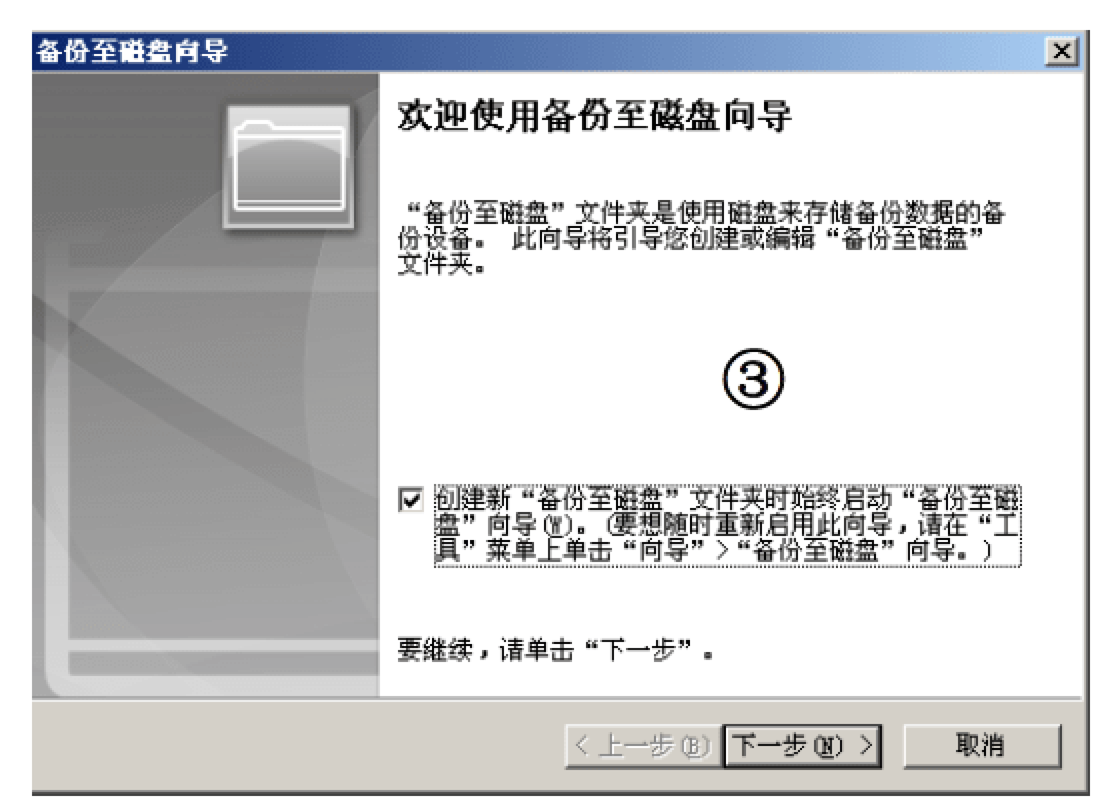
2. 在任何空白的地方右击，选择“新建‘备份至磁盘’文件夹”选项。



▲ 新建备份的文件夹

3. 此时会进入设备新建向导。单击“下一步”按钮继续。

4. 接下来键入这个备份文件夹的名称，就取名 NFS Backup。单击“下一步”按钮继续。



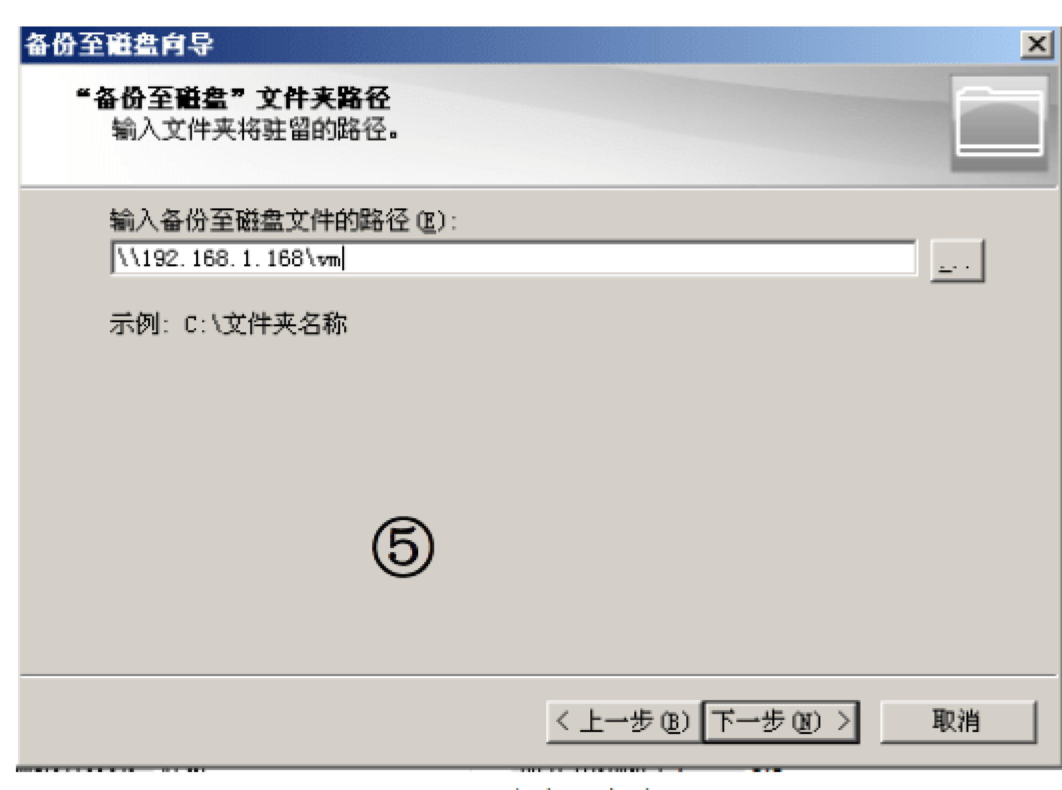
▲ 进入向导



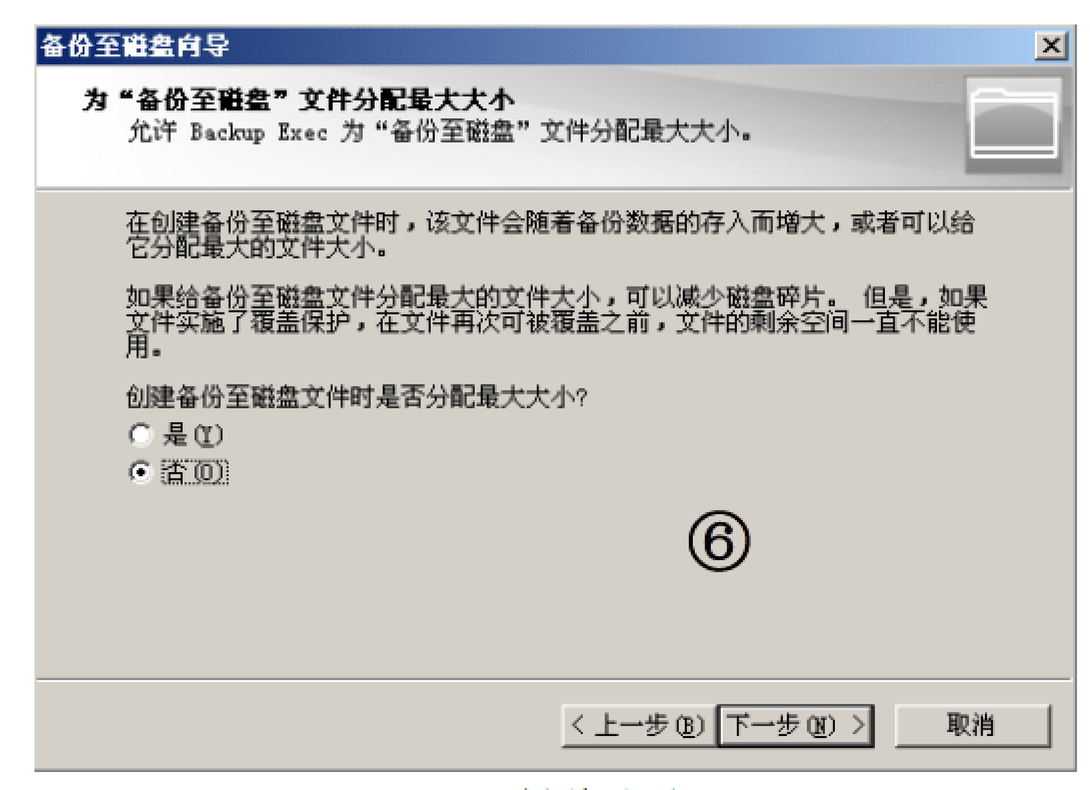
▲ 键入文件夹名称

5. 接下来选择 NFS 的路径。本例中 NFS 放在 192.168.1.223 的 VM 下，因此就键入这个文件夹名称。单击“下一步”按钮继续。

6. 此时会要求是否配置大小上限，我们选择不制约。单击“下一步”按钮继续。



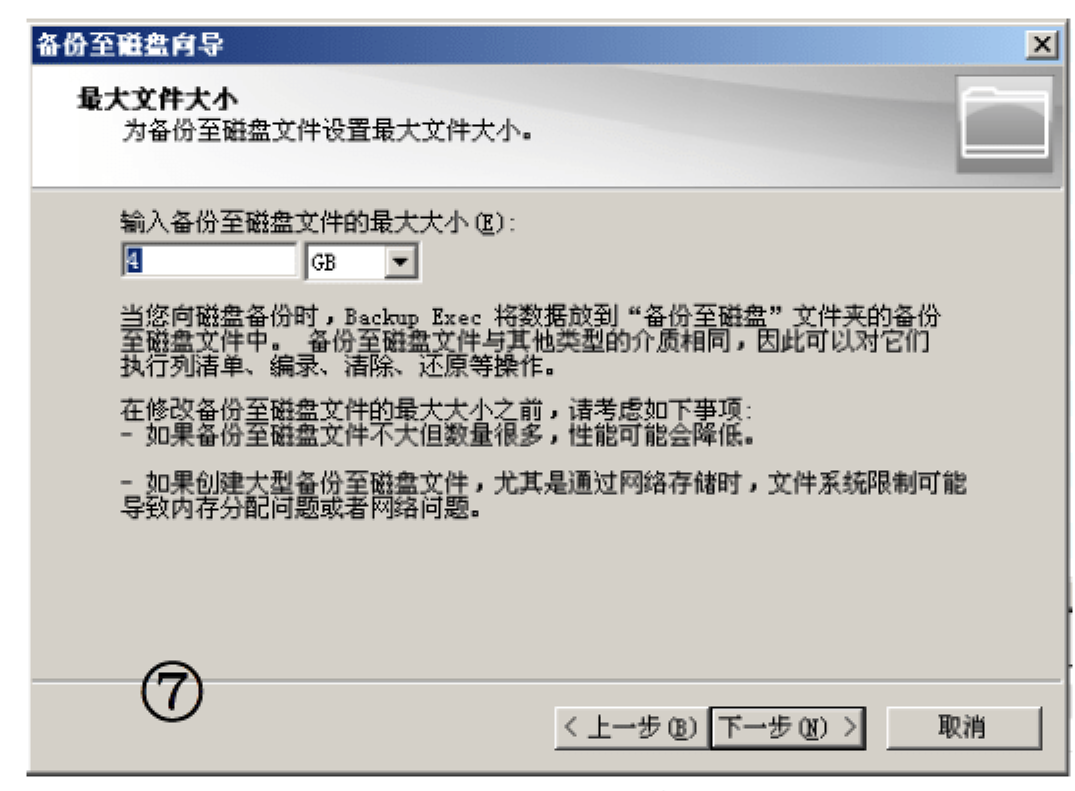
▲ 选择路径



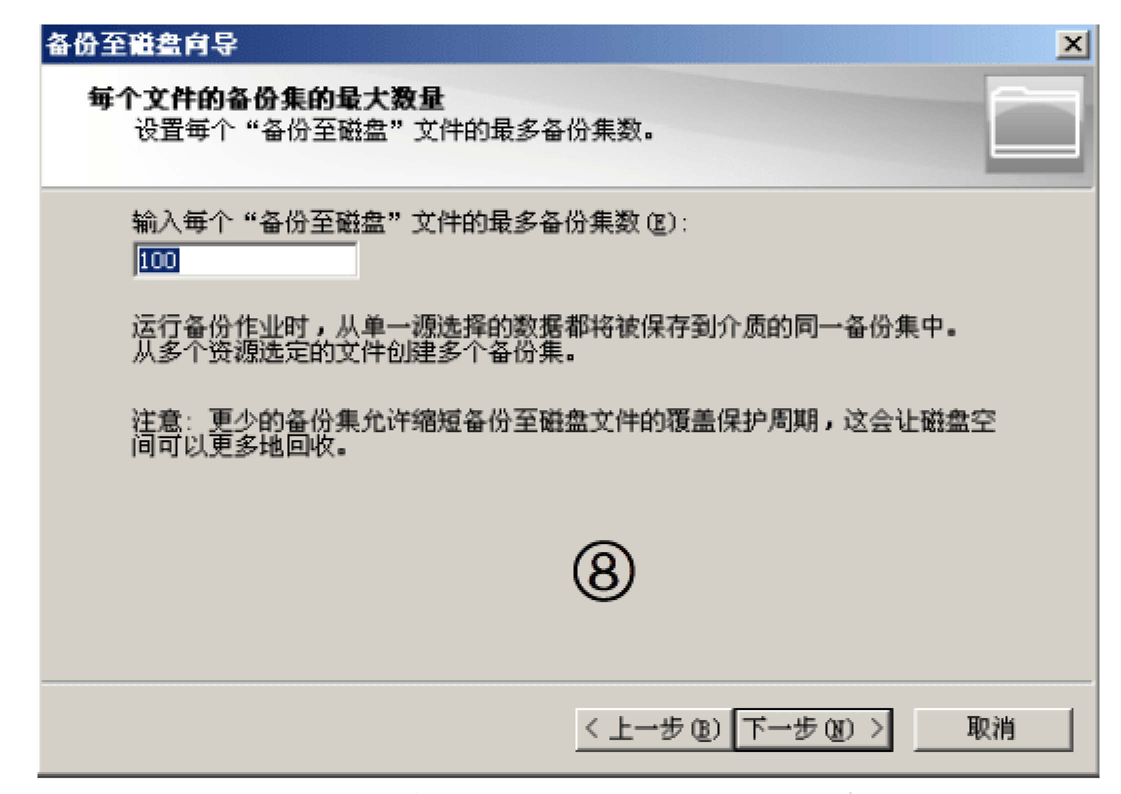
▲ 不制约大小

7. 接下来是配置文件的大小上限，而非真正的大小上限。我们就使用默认值，单击“下一步”按钮继续。

8. 接下来是备份集数，这个数字也不用管，单击“下一步”按钮继续。



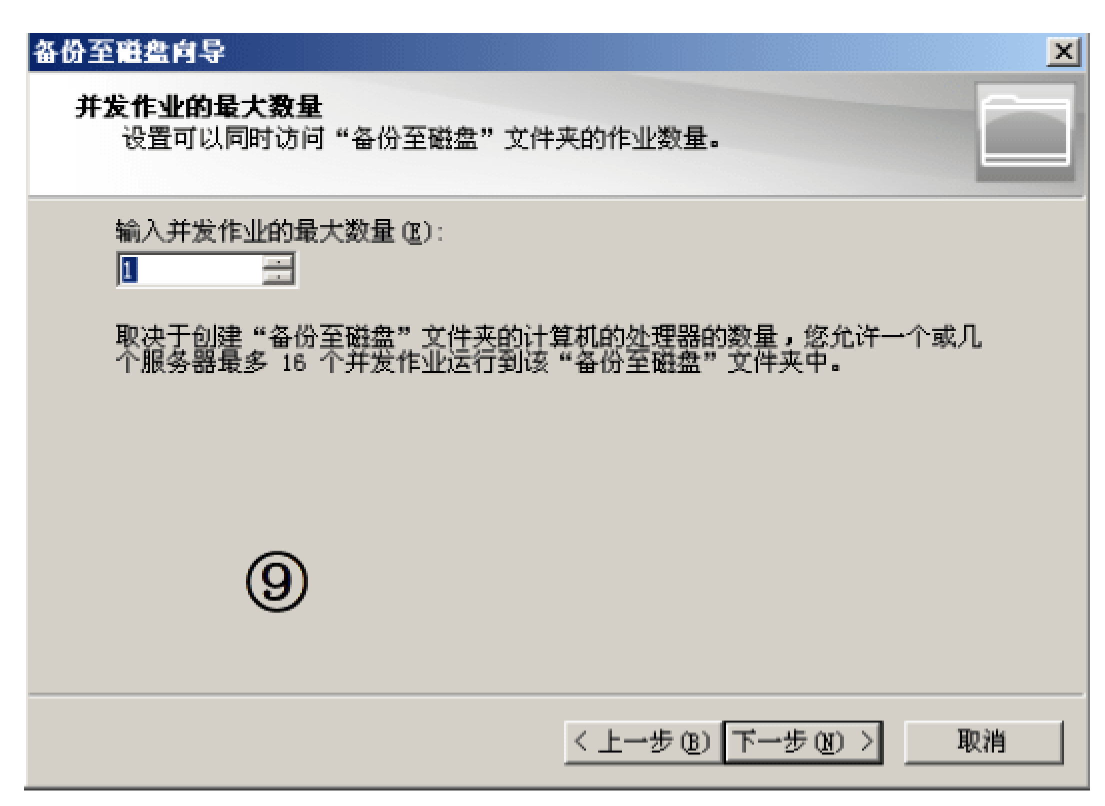
▲ 配置文件大小使用默认值



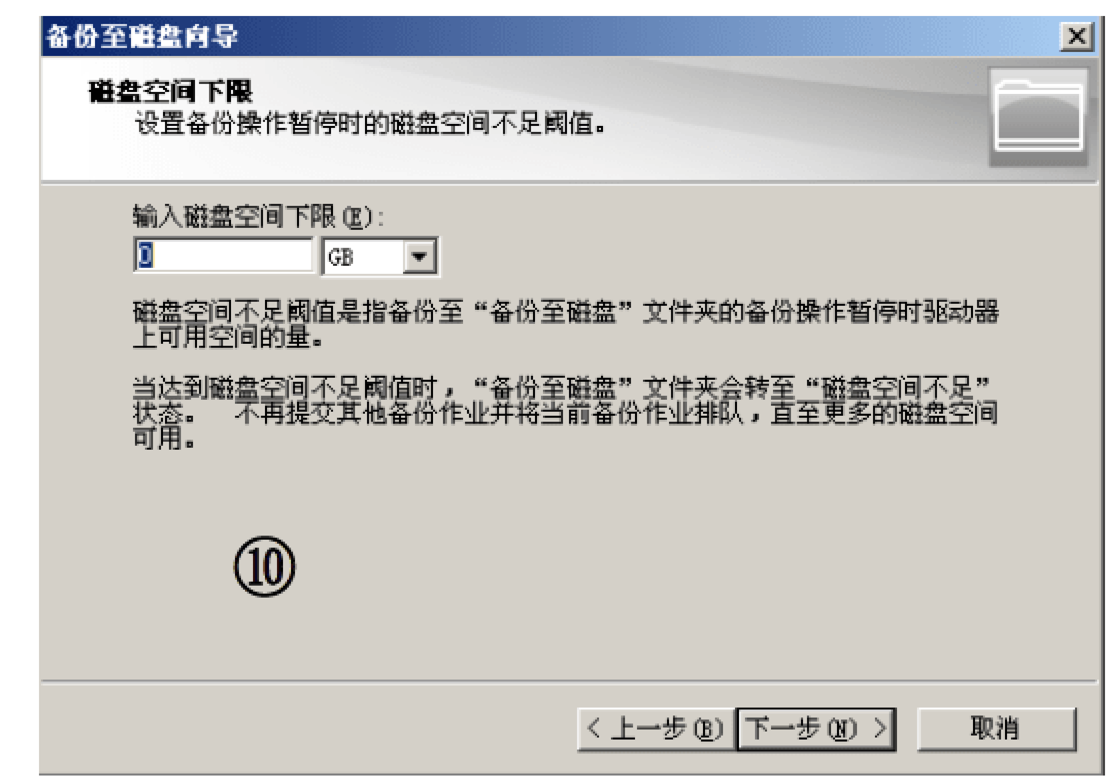
▲ 备份集数也不需要更动

9. 接下来是并发任务的最大数目，我们就使用默认值。单击“下一步”按钮继续。

10. 接下来是键入磁盘空间不足的临界值，我们也使用默认值。单击“下一步”按钮继续。



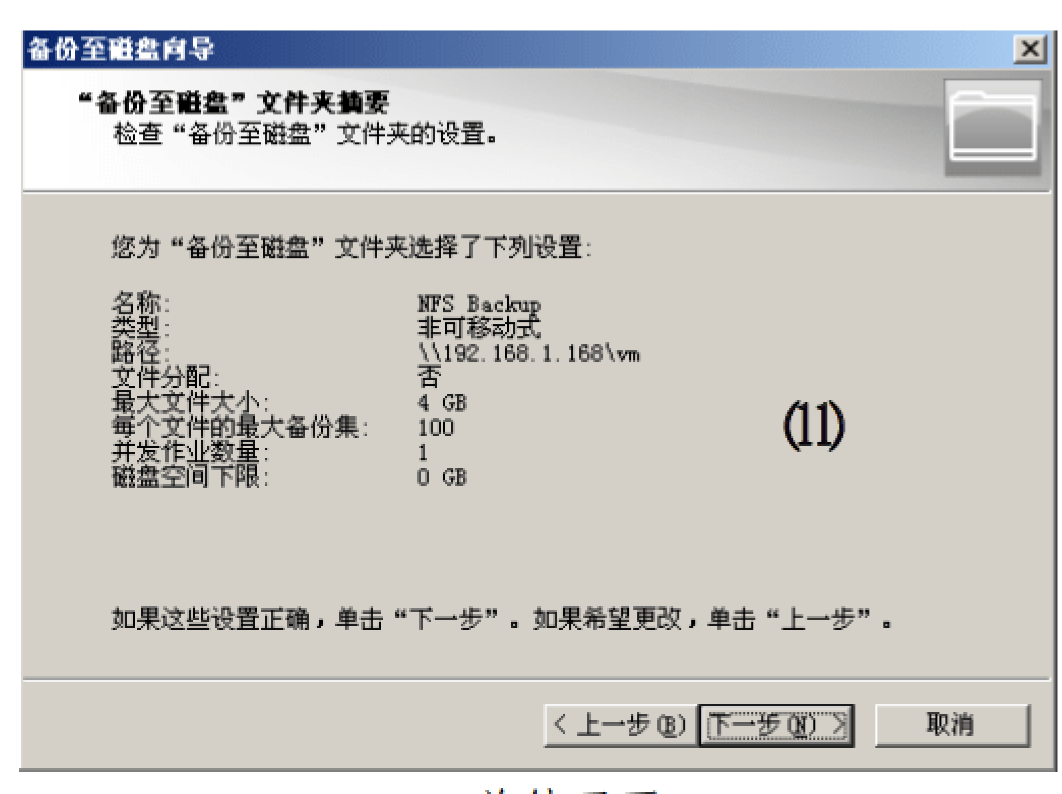
▲ 并发任务数目也不需要改动



▲ 这里也使用默认值

11. 最后是总结画面，单击“下一步”按钮继续。

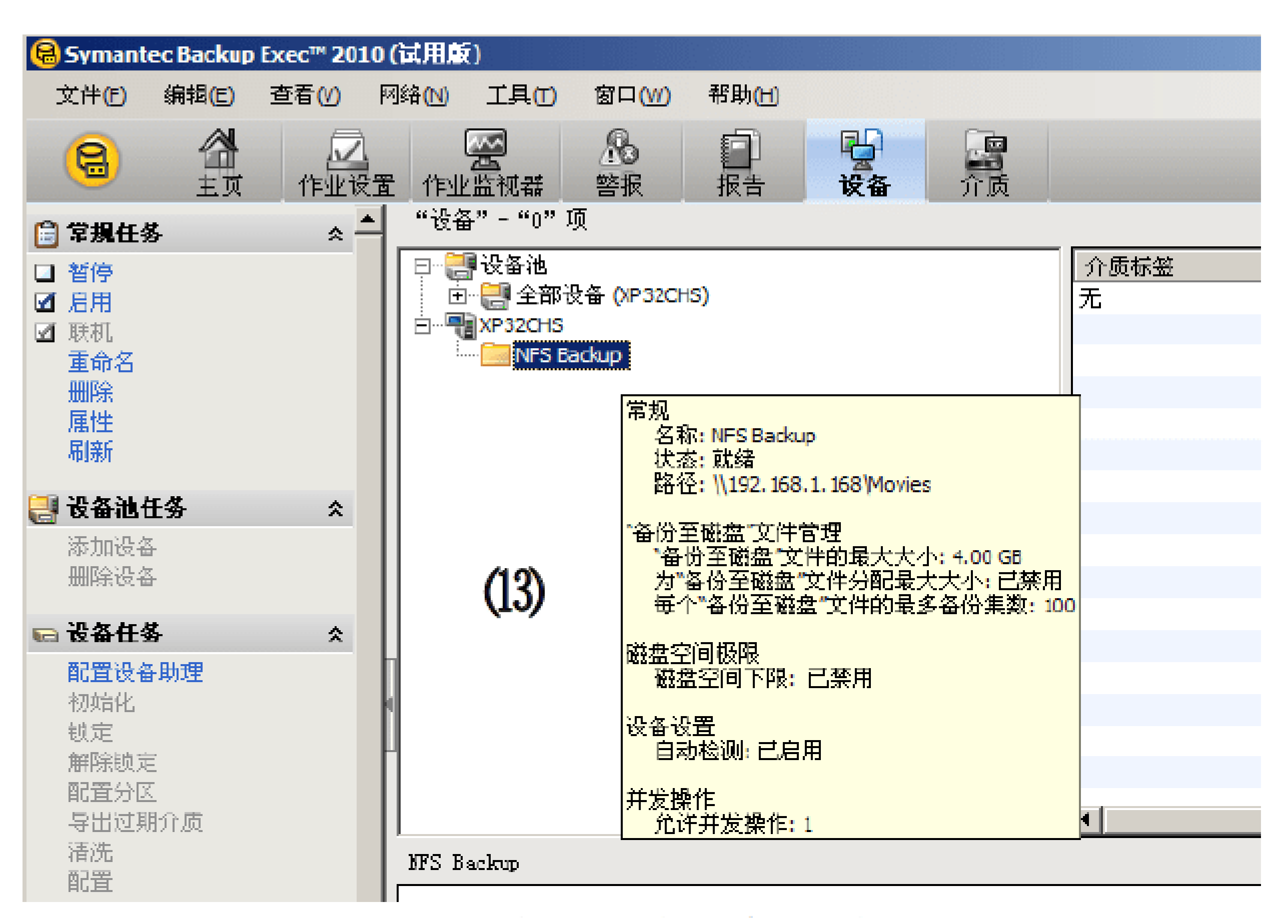
12. 接下来单击“完成”按钮即创建落实。



▲ 总结画面

▲ 开始创建

13. 之后可以在系统主画面中看到这个已经创建好的备份设备。



▲ 该备份设备已创建落实

20.3.2 使用 Symantec Backup Exec 2010 备份及还原 VM

Symantec Backup Exec 2010 的使用相当简单，只要你有任何使用备份软件的经验，就会发现备份还原 VM 和备份一般的数据大同小异。我们就来看看 Symantec Backup Exec 2010 如何备份及还原 VM。

1. 使用 Symantec Backup Exec 2010 备份 VM

当安装好 Symantec Backup Exec 2010 之后，别忘了将 VCB 也安装起来，之后就可以进行备份了。Symantec Backup Exec 2010 备份 VM 必须先将要备份的 VM 加入，我们就来看看如何用 Symantec Backup Exec 2010 备份 VM。

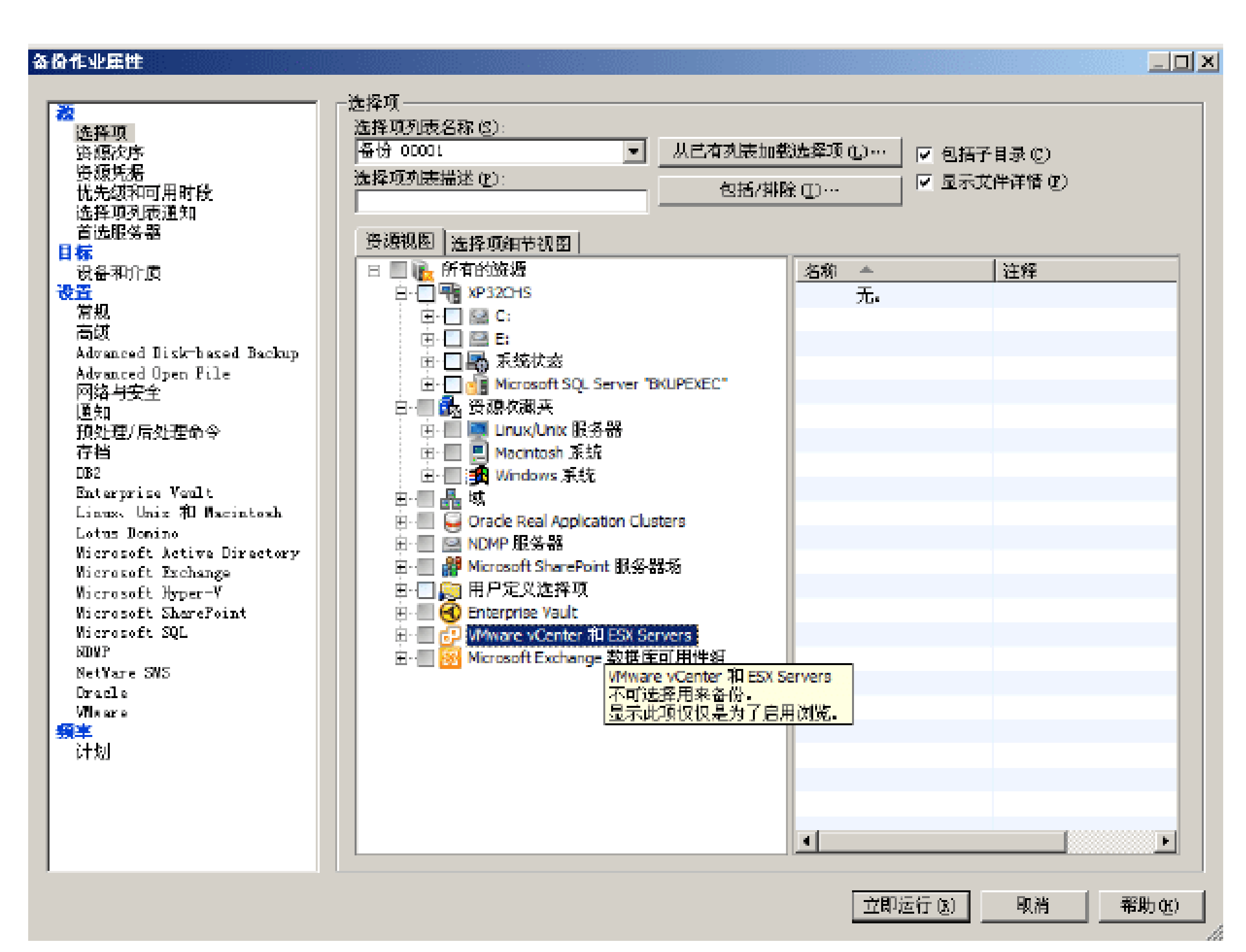
▶ 使用 Symantec Backup Exec 2010 备份 VM

1. 单击系统主画面中的黄色图示，并且选择“新建备份任务”选项。



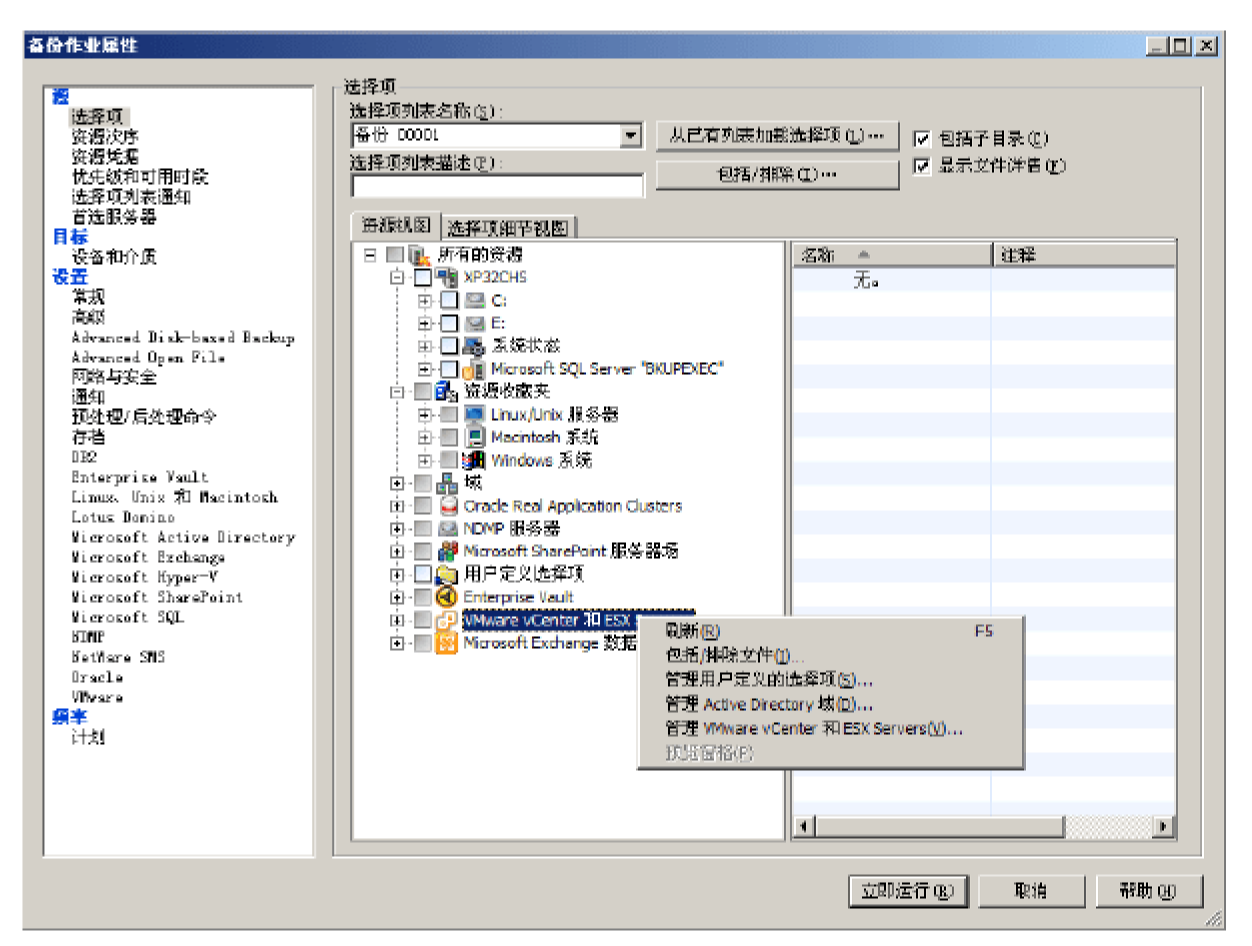
▲ 选择新建备份任务

2. 此时会进入选取备份数据的画面。进入 VMware vCenter 和 ESX 服务器，还无法备份，因为我们还没有加入 vCenter 的服务器。



▲ 当前还无法备份，因为没有加入 VM

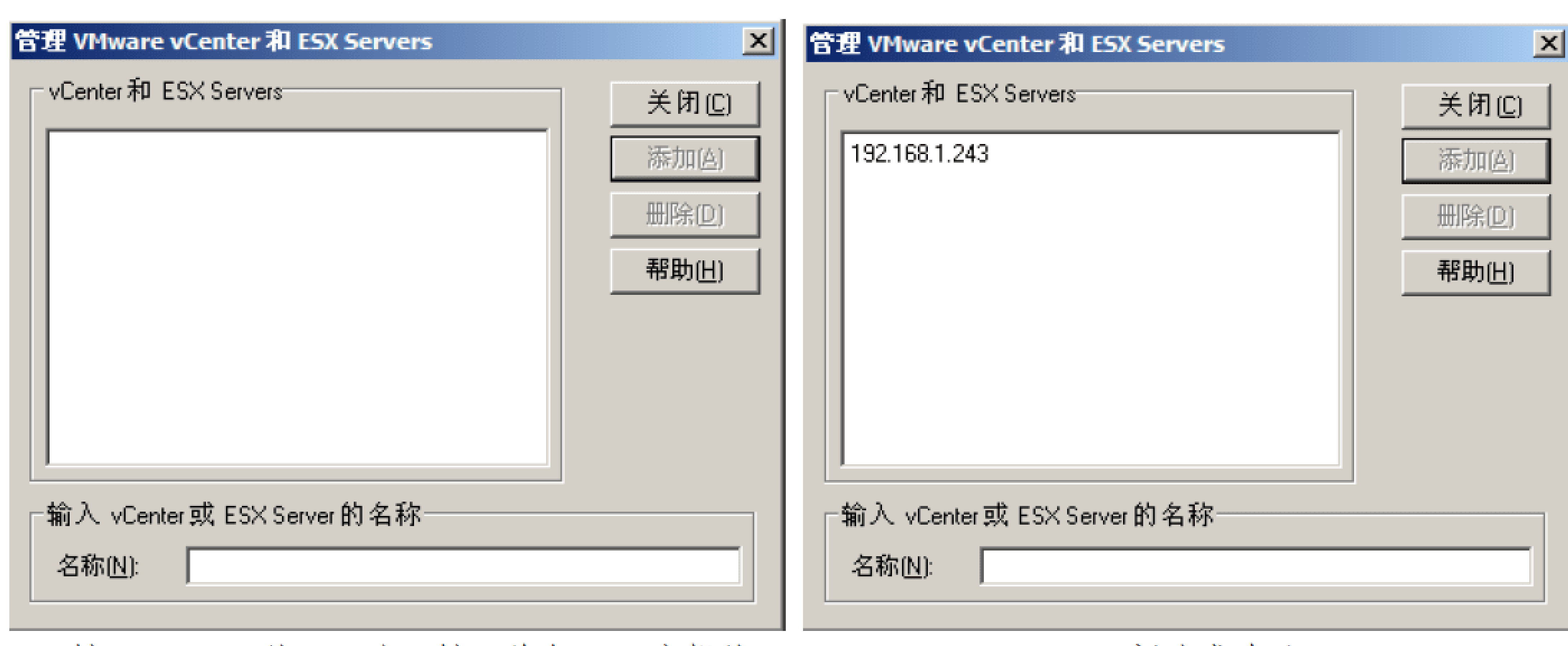
3. 此时我们在“VMware vCenter 和 ESX 服务器”上右击，选择“管理 VMware vCenter 和 ESX 服务器”选项。



▲ 选择管理 VMware vCenter 和 ESX

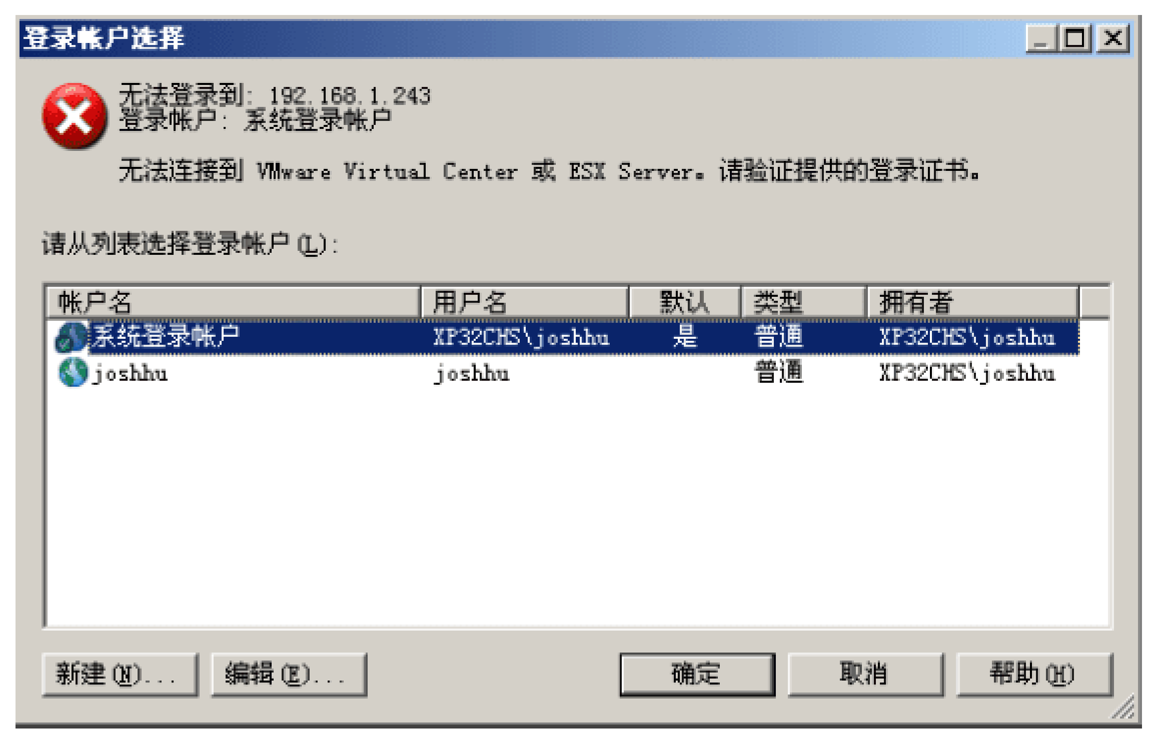
4. 在弹出的窗口中，键入 vCenter Server 的 IP，并单击“新建”按钮。

5. 此时会弹出 vCenter Server 的 IP，单击“关闭”按钮结束。



▲ 键入 vCenter 的 IP，也可键入单个 ESX 主机的 IP　　▲ 新建成功了

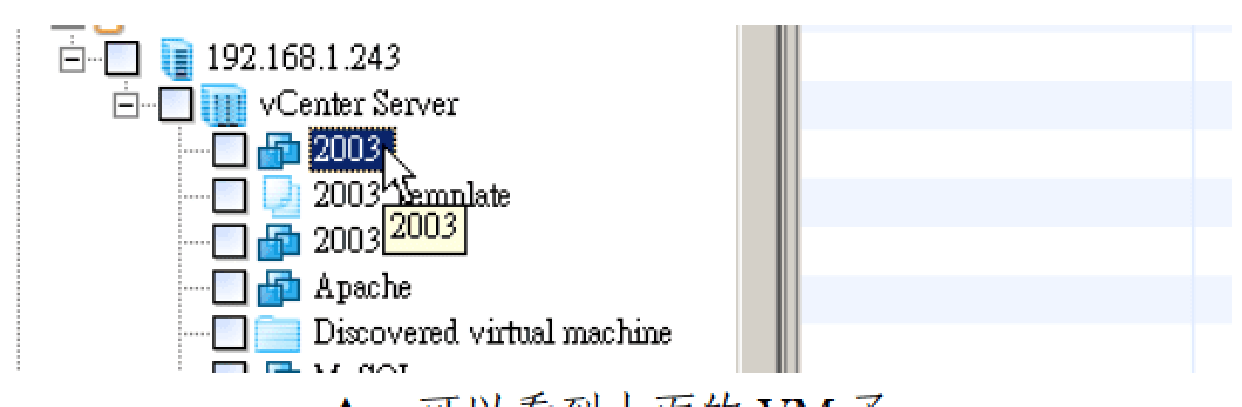
6. 单击该服务器，会弹出选取用户的窗口，我们选择刚才创建的用户名。单击“确定”按钮。



▲ 此时系统会要求我们选择登录该 vCenter 的账号

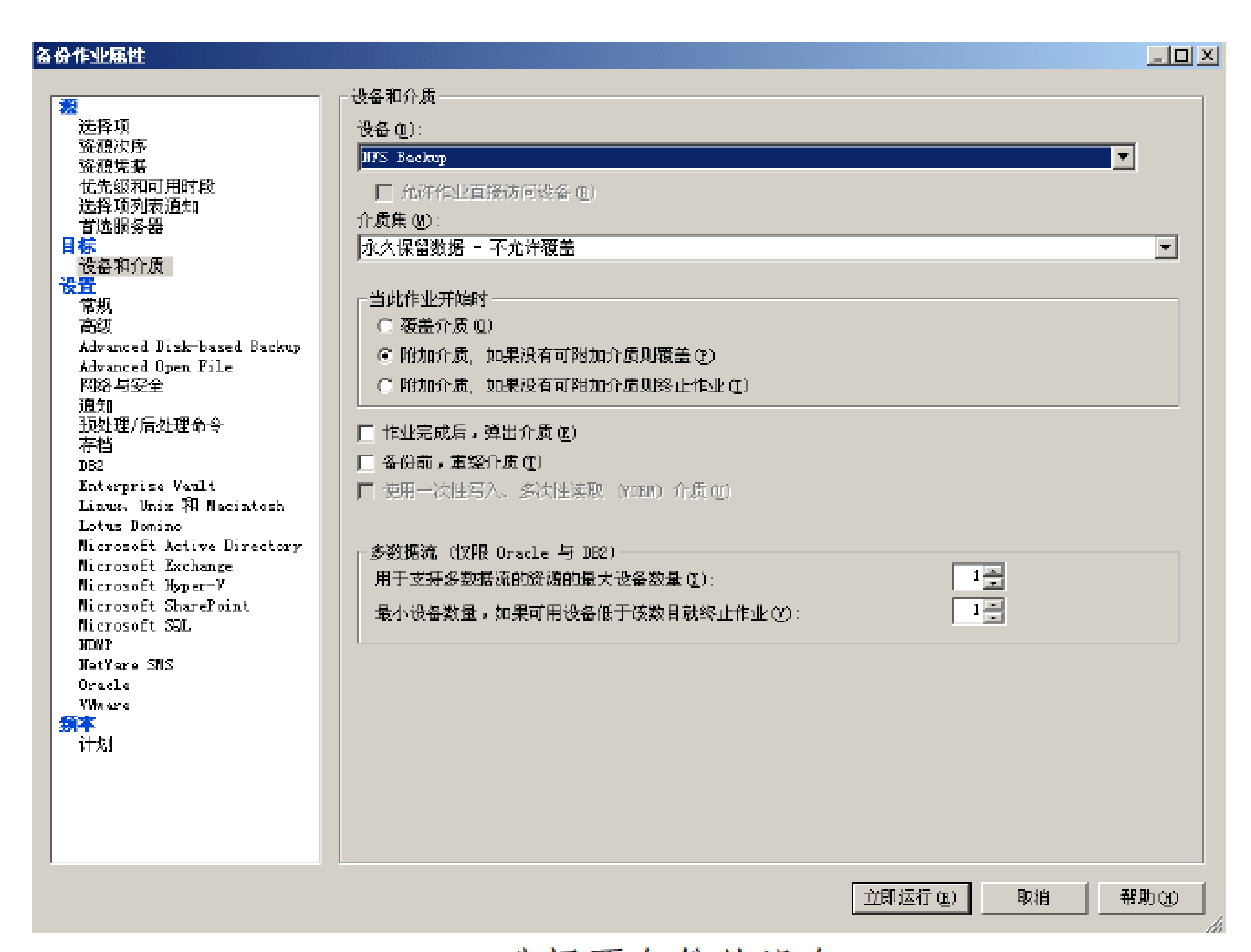
7. 此时该 vCenter 上所有的 VM 都会弹出了。选择任何一个 VM，都可以看到该 VM 下

拥有的文件。此时我们选择 Nostalgia 这个 VM 来备份。选中这个 VM，就会将其所有文件选取。



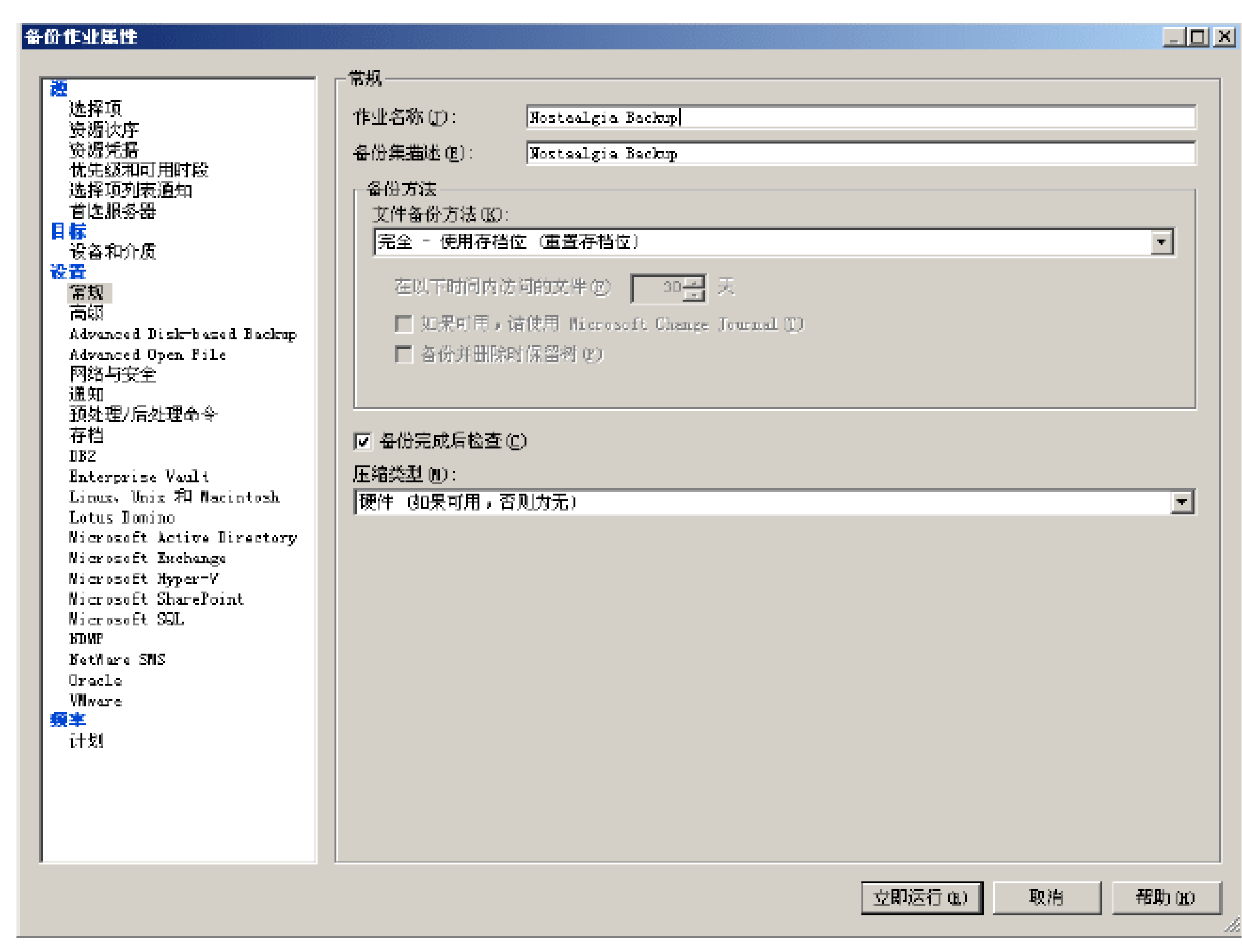
▲ 可以看到上面的 VM 了

8. 接下来选择设备与介质，选择要备份的目的。我们就选择刚才创建的 NFS Backup。



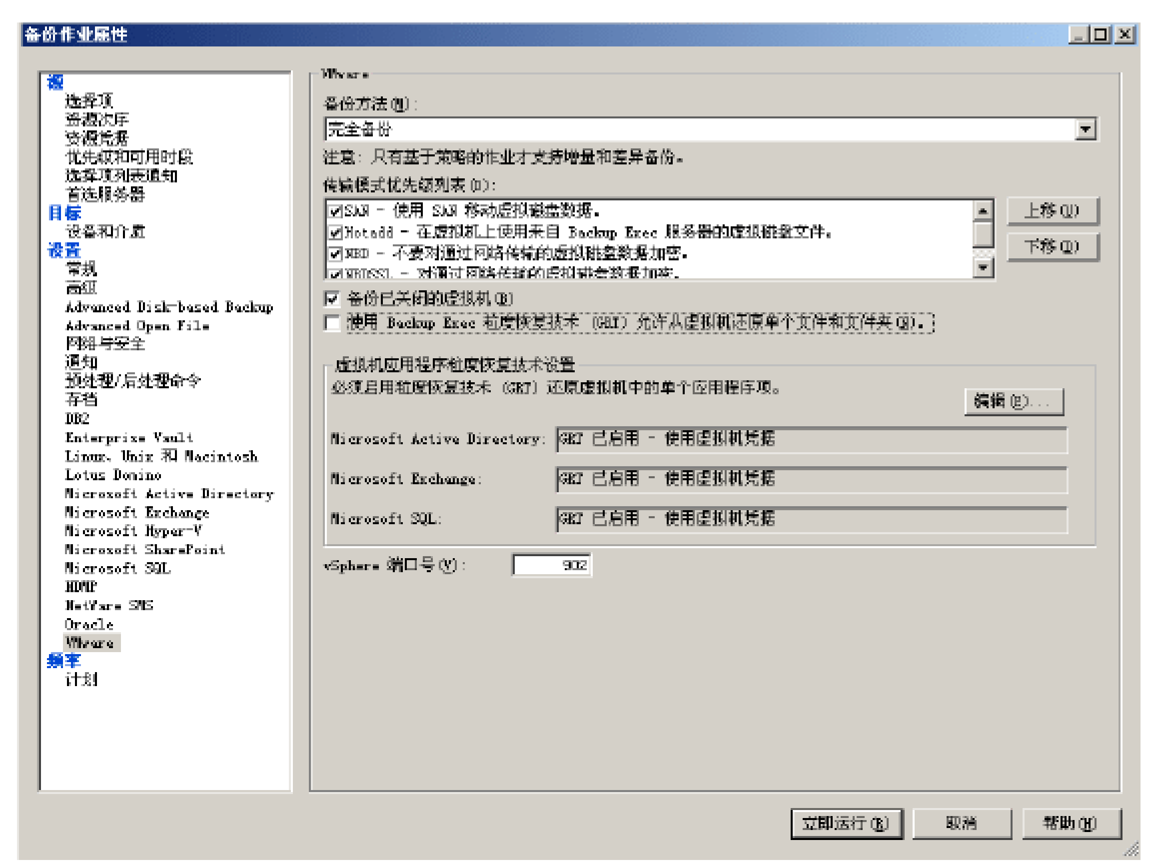
▲ 选择要备份的设备

9. 接下来在“常规”选项卡，我们将这个备份名命名为 Nostalgia Backup，并且选择完整备份。



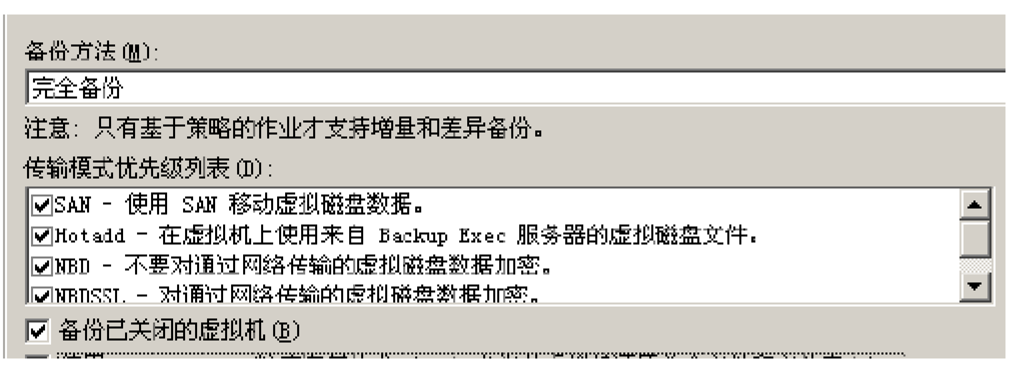
▲ 为这个备份任务命名

10. 接下来进入 VMware 选项卡。将备份手段选择为完整，由于我们的 VM 并没有安装附加程序，因此将 GRT 选项关闭。



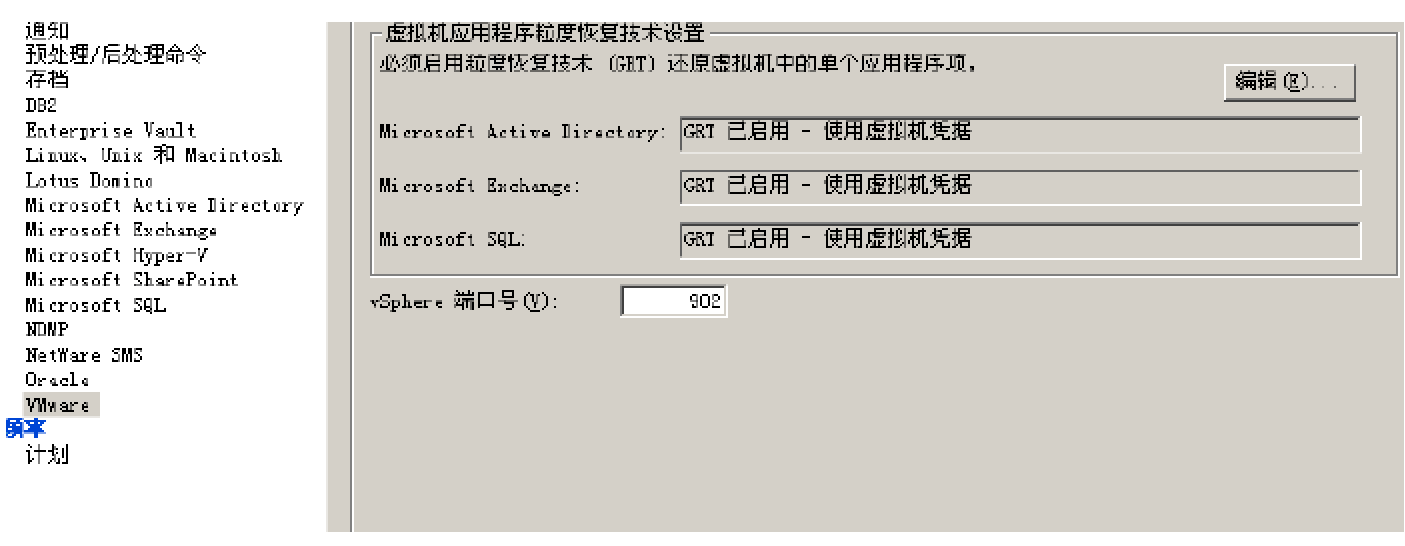
▲ 别忘了将 GRT 关闭

11. 在“传输模式优先级列表”中，我们将 NBD 移到最上面，因为这是以网络为主的备份。



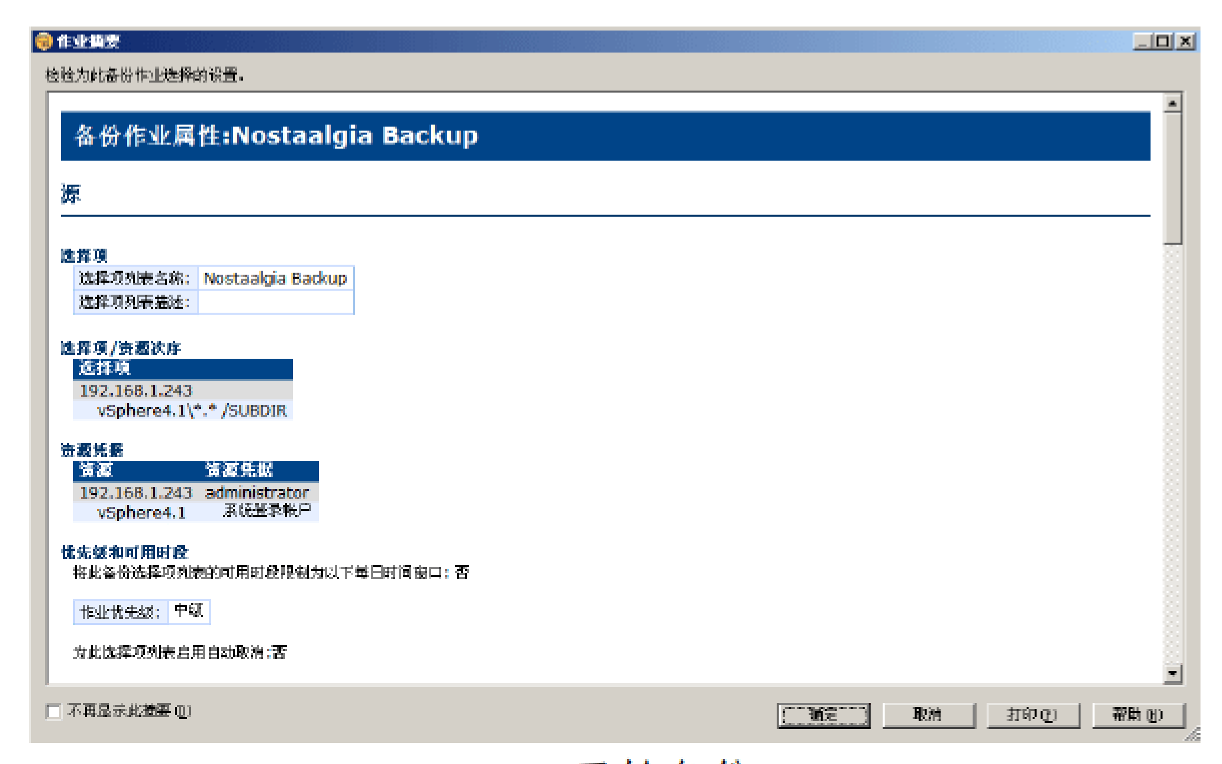
▲ 如果你的 Backup Proxy 无法连到独立存储设备，就得选择 NBD 模式

12. 接下来我们直接单击“立即运行”按钮。



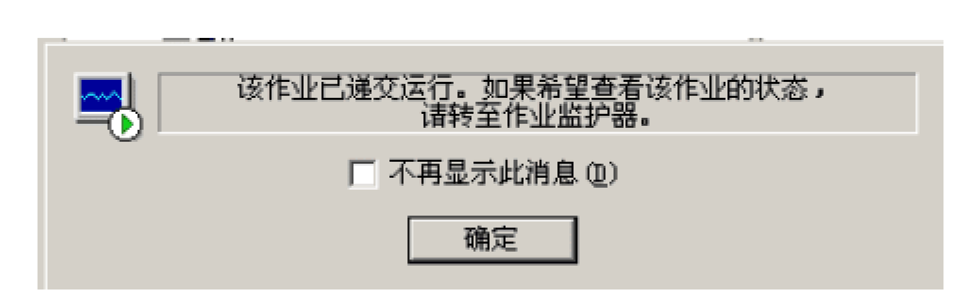
▲ 立即运行

13. 此时会弹出总结画面，单击“确定”按钮即可开始备份。



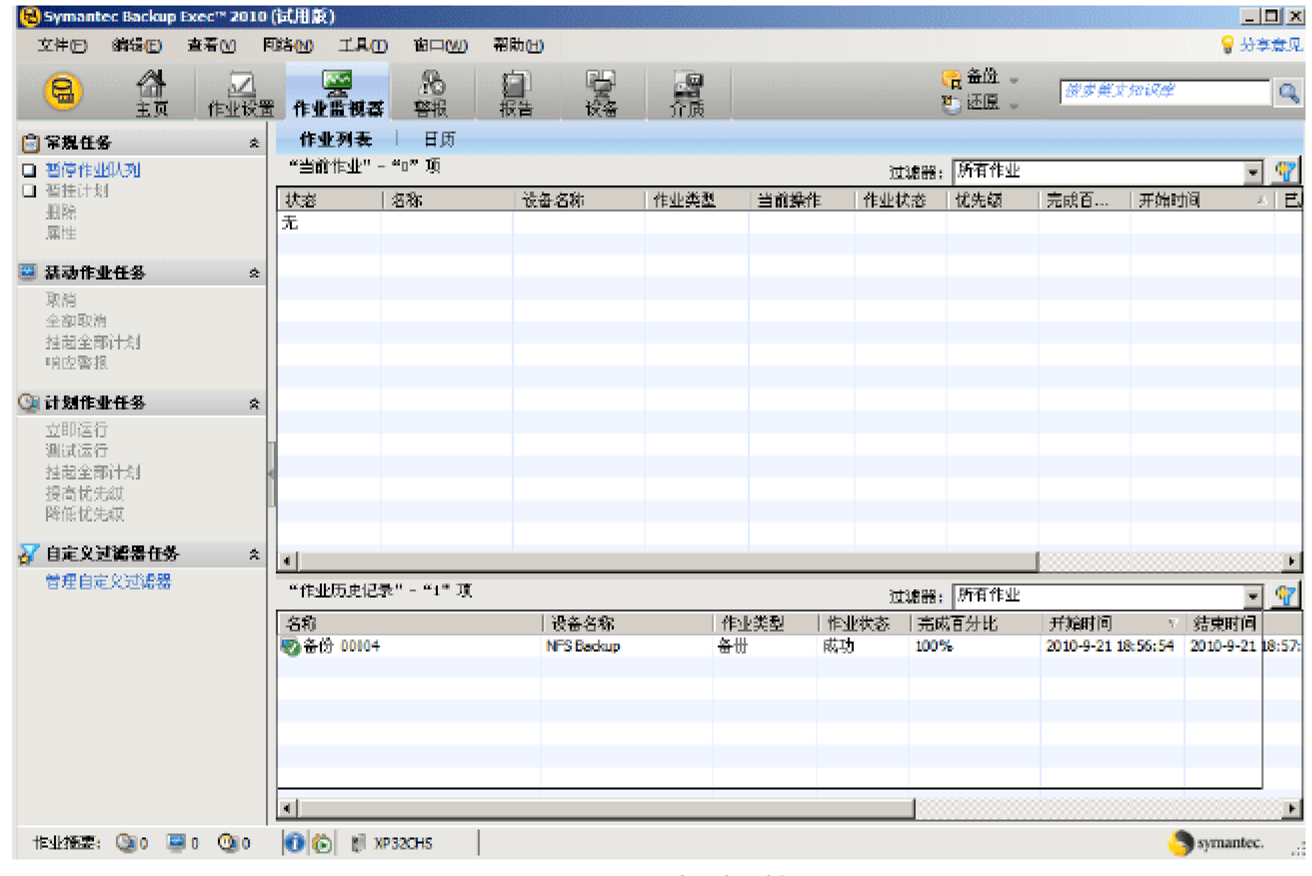
▲ 开始备份

14. 在画面中单击“确定”按钮开始备份。



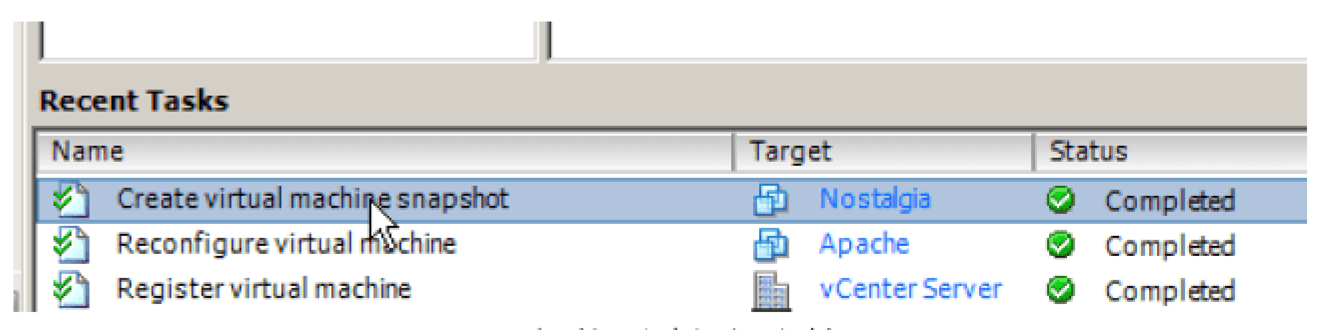
▲ 开始备份

15. 选择“任务监视器”选项卡，可以看到备份正在进行，而当前的任务是创建快照。



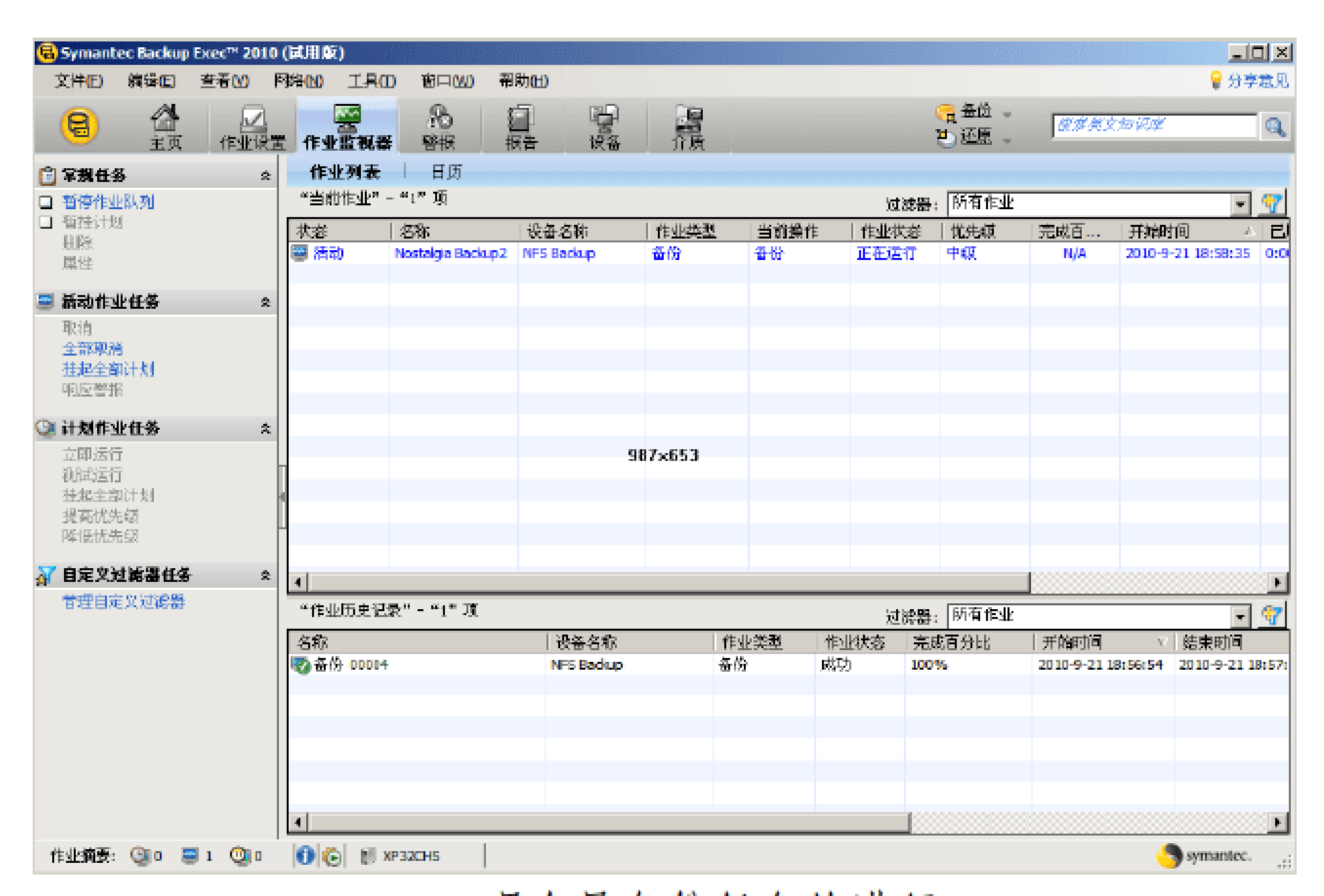
▲ 正在备份

16. 此时我们可以在 vCenter 中看到该 VM 拥有一个新的快照，这就是 Backup Exec 创建的快照。



▲ 备份时创建的快照

17. 接下来就会开始备份。当前的任务就是备份，会花上较长的时间。



▲ 现在是备份任务的进行

18. 当弹出画面时，意味着备份落实了。



▲ 备份落实，但由于 VM 没有安装 Agent，会有异常生成，但备份已成功

2. 使用 Symantec Backup Exec 2010 还原 VM

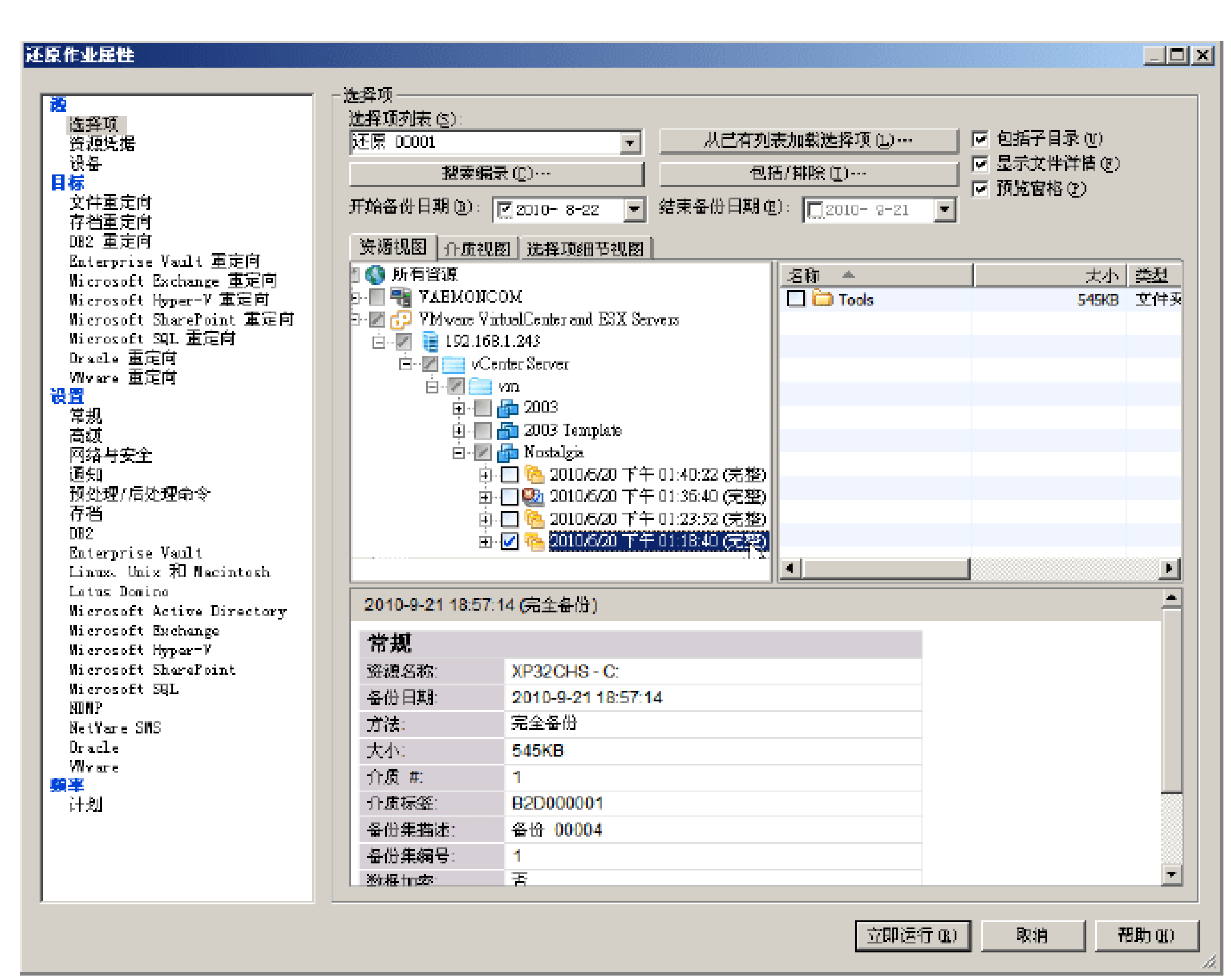
当我们创建了备份之后，在紧急情况下需要还原 VM。在 Symantec Backup Exec 2010 下还原 VM 再简单不过了！只要遵守其还原的步骤，在简单几个动作之后，VM 就可以还原到原来的 ESX 主机上，我们就来看看还原的完整步骤。

▶ 在 Symantec Backup Exec 2010 下还原 VM

1. 进入 Symantec Backup Exec 2010，并且选择“新建还原作业”选项。
2. 此时从 VM 中选择刚才备份的 Nostalgia。要注意的是，在还原这个 VM 之前，我们必须确定 ESX 上并没有这个 VM。

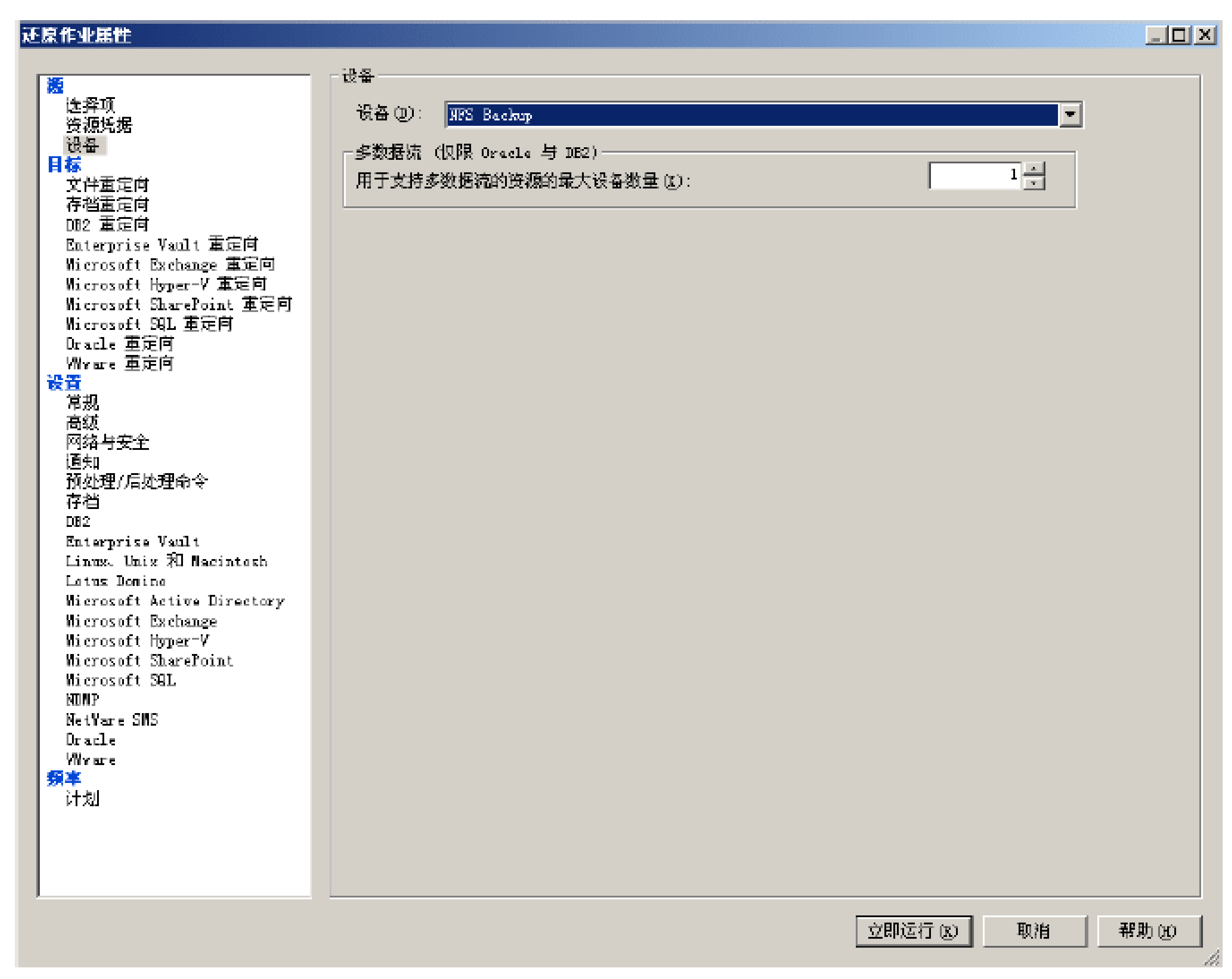


▲ 新建还原任务



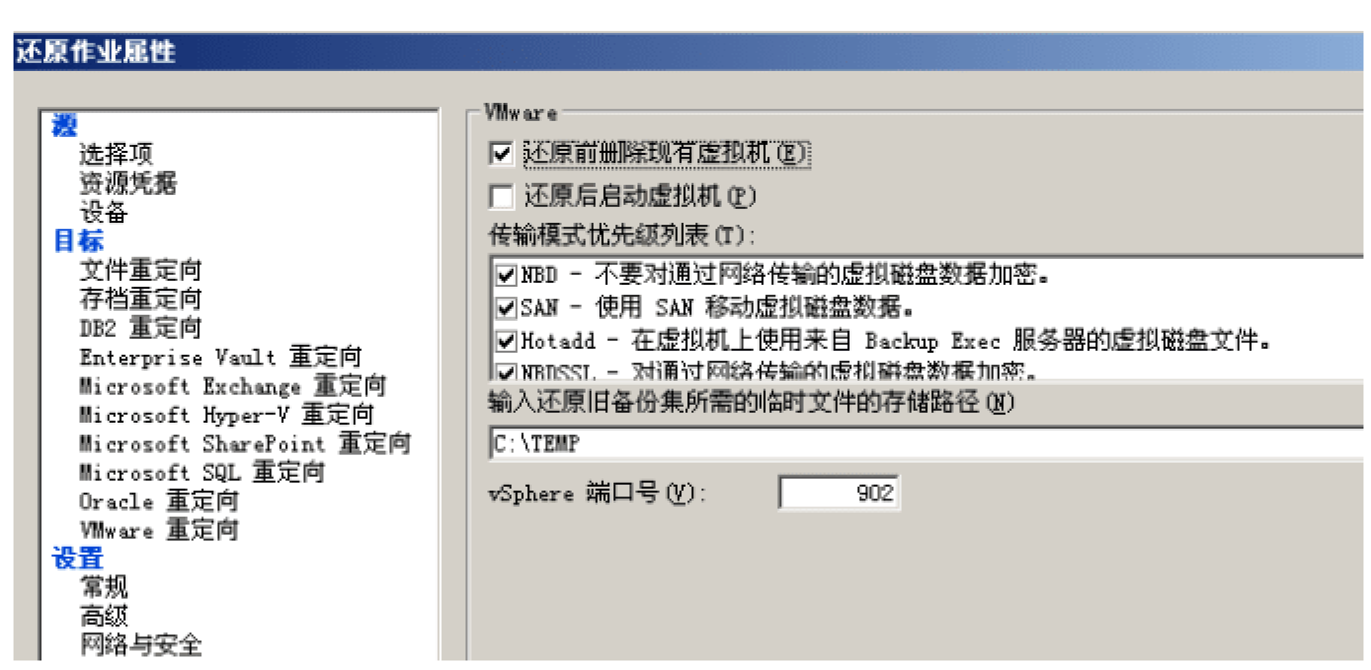
▲ 选择刚才备份的图像

3. 接着在“设备”下拉列表框中选择刚才备份的目的地，如 NFS Backup。



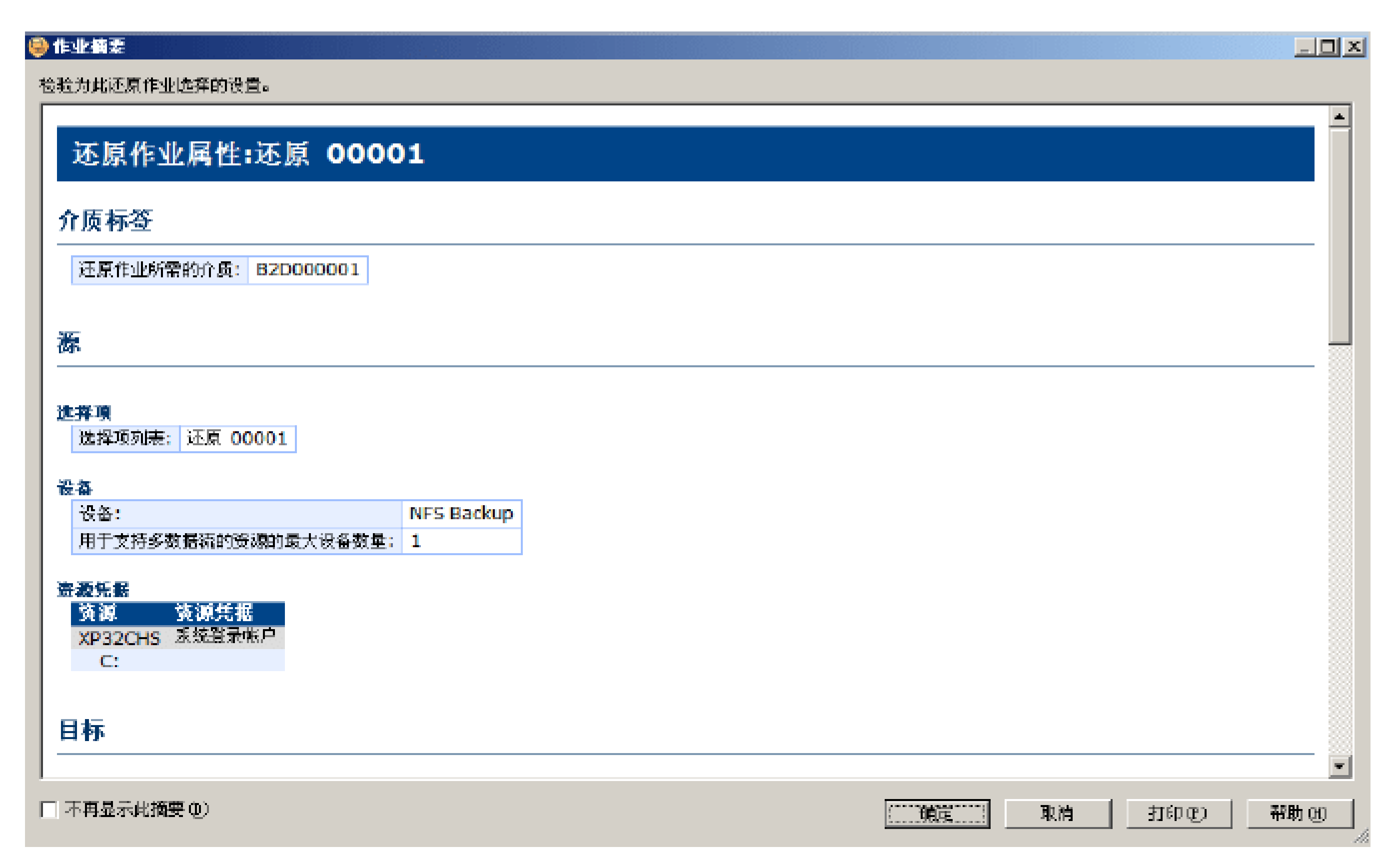
▲ 选择刚才备份的目的地

4. 接下来在 VMware 选项卡中，勾选“还原前删除现有虚拟机”复选框。要不然会弹出错误消息。接下来直接单击“立即运行”按钮。



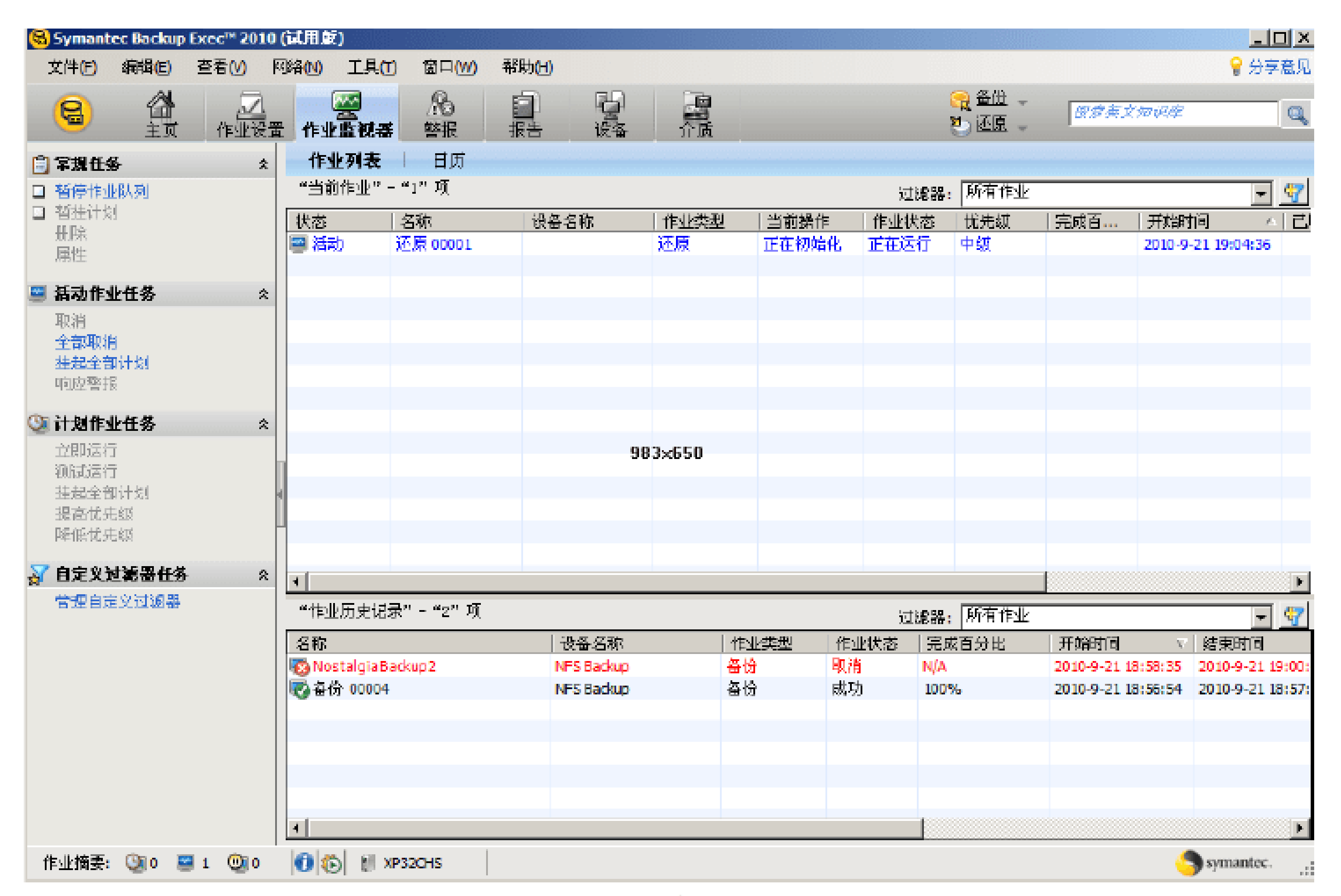
▲ 别忘了要先将 ESX 上的 VM 删除

5. 此时弹出总结画面，单击“确定”按钮。



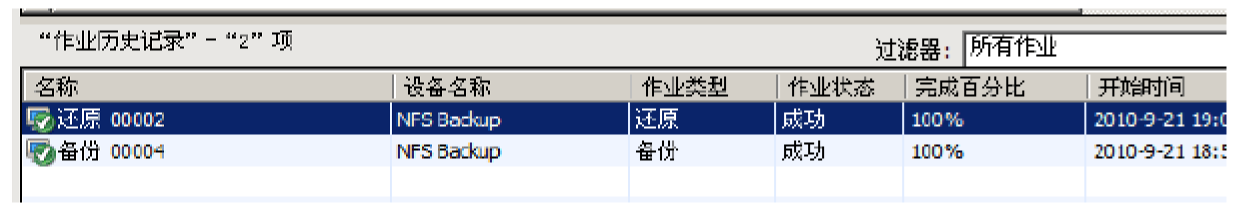
▲ 总结画面

6. 接下来会进行还原。



▲ 开始还原

7. 当弹出图中的画面时，意味着还原落实了。



“作业历史记录” – “2” 项　　过滤器：所有作业

名称	设备名称	作业类型	作业状态	完成百分比	开始时间
还原 00002	NFS Backup	还原	成功	100%	2010-9-21 19:(
备份 00004	NFS Backup	备份	成功	100%	2010-9-21 18:5

▲ 还原落实



▲ 可以在任务日志上看到还原的表项

注意

什么是 GPT

GPT 是让你的 VM 能精确到文件等级的备份/还原，但前提是 VM 必须安装 Symantec Backup Exec Remote Agent。虽然会让备份/还原的功能加强，但在每一个 VM 中安装不同的 Agent 会添加具体及管理上的成本。读者们可以视业务的需求来决定是否要安装这个 Agent，但就大部的情况来说，备份整个 VM 之后再做增量备份已经足够了。

结 语

VCB 是 vSphere 备份的基础，IT 人员不需要学习更多的备份技术，以当前最熟悉的备份软件即可进行 VM 的备份。再配合第三方软件，进行 VM 的备份还原更加方便了。然而在 vSphere 的高级版本中，VMware 推出了更高级的 VMware Data Recovery，使用 Virtual Appliance 的方式来进行备份及还原，我们会在下一章中对 Virtual Appliance 及 VDR 这些功能有更完整的介绍，但大家还是要打好 VCB 的备份基础，你辛苦架设的 vSphere 环境才能有最基础的保障。

读书笔记

第 21 章 使用 VMware Data Recovery

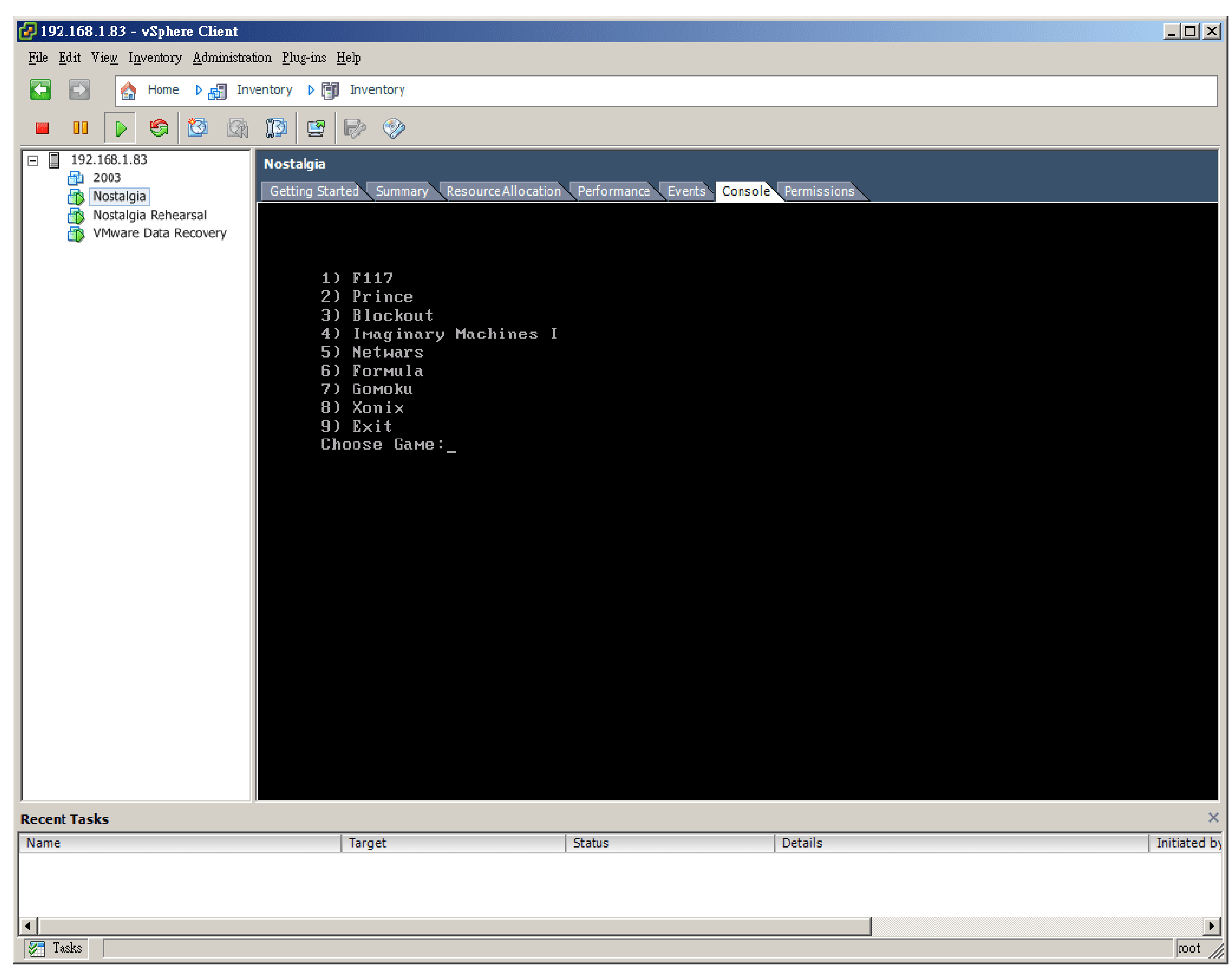
关键词：

- VMware 使用 VA 帮助软件厂家交付服务
- VA 使用标准的 OVF 格式，VMware 是创始人之一
- vSphere 提供了许多现成的 VA 使用
- 高级版的 vSphere 使用 VDR 进行备份还原
- VDR 可完整备份及增量备份
- VDR 有一演练还原的功能
- 还原时可选择任意时间点

VMware 除了提供虚拟环境之外，也是一个软件服务的供货商，主要的功能还是提供虚拟环境中的服务。而 Virtual Appliance 则是其提供的方式。有了标准 OVF 格式的 VA 之后，软件的发布将更加方便也更有弹性，而 VMware 本身的 VDR 就是很好的例子。有了 VDR 之后，VM 的安全性获取更大的保障，备份还原 VM 的任务也更加轻松。

21.1 理解 Virtual Appliance

我们都知道在 vSphere 中，一个 VM 就是一个独立的操作系统，这个 VM 中可以运行我们需要的服务。由于 VM 拥有各种比物理机强大的功能，因此除了原来物理机中的服务之外，VM 也可以提供操作系统之上的服务，如应用软件。有了这个基础，许多应用软件厂家会利用一些免费的 Linux，并且在其上架设各式各样的软件，并且将整个服务加壳成一个 VM，成为产品出售或出租，这种以 VM 为单位的产品，当前已经成为虚拟世界中最流行的服务了。



▲ 通常一个 VM 被视为一个操作系统，但其上也可以有服务

21.1.1 什么是 VMware 的 VA

Virtual Appliance（简称 VA）是 vSphere 平台中很重要的一个部件。在 vSphere 中称为 vAPP。vSphere 的 VA 就是一个 vSphere 的图像文件。VA 最大的优点，就是不需要安装、维护及配置，只要将 VA 挂上，在 vSphere 中引导，这个 VA 就立即可用了。

1. VA 不是单纯的 VM

大家一定要知道，VA 并不是 VM，更像是一个软件的图像文件，包括了多层次的软件运行架构，并且在 VM 中运行。就像物理机一样，VM 对 VA 来说，只是一个运行的平台，提供 VA 一个运行的环境，VM 本身是并不包括任何服务的软件的。

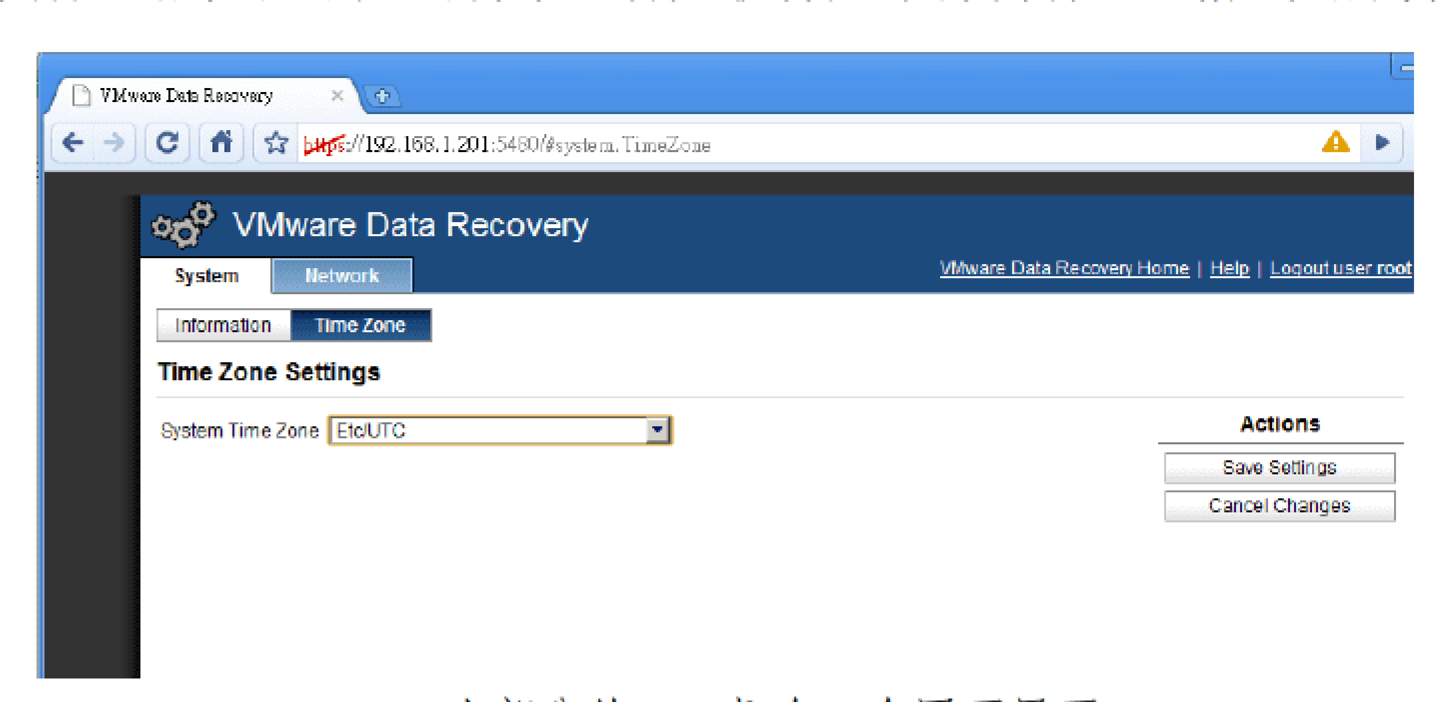
Nostalgia Rehearsal
VMware Data Recovery

```
   Vendor: VMware    Model: Virtual disk      Rev: 1.0
   Type:   Direct-Access                      ANSI SCSI revision: 02
SCSI device sdc: 209716 512-byte hdwr sectors (107 MB)
sdc: Write Protect is off
sdc: cache data unavailable
sdc: assuming drive cache: write through
SCSI device sdc: 209716 512-byte hdwr sectors (107 MB)
sdc: Write Protect is off
sdc: cache data unavailable
sdc: assuming drive cache: write through
sd 0:0:2:0: Attached scsi disk sdc
sd 0:0:2:0: Attached scsi generic sg2 type 0
   Vendor: VMware    Model: Virtual disk      Rev: 1.0
   Type:   Direct-Access                      ANSI SCSI revision: 02
SCSI device sdc: 209716 512-byte hdwr sectors (107 MB)
sdc: Write Protect is off
sdc: cache data unavailable
sdc: assuming drive cache: write through
SCSI device sdc: 209716 512-byte hdwr sectors (107 MB)
sdc: Write Protect is off
sdc: cache data unavailable
sdc: assuming drive cache: write through
sd 0:0:2:0: Attached scsi disk sdc
sd 0:0:2:0: Attached scsi generic sg2 type 0
_
```

▲ 提供服务的 VM 就可称为 VA

2. VA 的性质

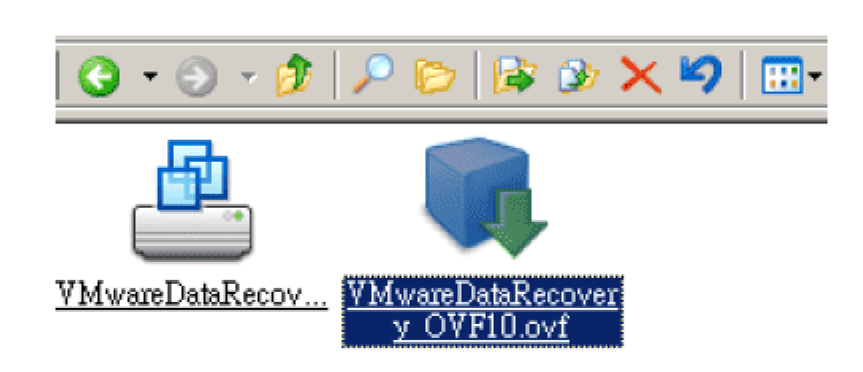
大部分的 VA 一定有提供一个网页的用户界面来落实该 VA 服务的配置。包括该 VA 的网络及软件功能。当前大部分的软件厂家都有提供 vSphere 格式的 VA，而 VMware 自身的产品也是用 VA 的方式来发行，相信在不久的将来，各大软件厂家都会有 VA 格式的服务。



▲ 大部分的 VA 都有一个网页界面

21.1.2　Virtual Appliance 最常用的格式：OVF

VA 既然是图像文件，就必须有一个公用的格式。在 2007 年 9 月，Dell、HP、IBM、微软、VMware 及 XenSource 等公司提出了开放虚拟机格式（Open Virtual Machine Format），后来改成 Open Virtualization Format 简称为 OVF。OVF 是一个开放的标准，主要的目的就是用来加壳及提交 VA，让软件服务能在大部分主流 Hypervisor 平台的 VM 上运行，而作为 OVF 的创始厂家之一，vSphere 的主要 VA 格式当然就是 OVF。



▲ 大部分的 VA 都是 OVF 格式

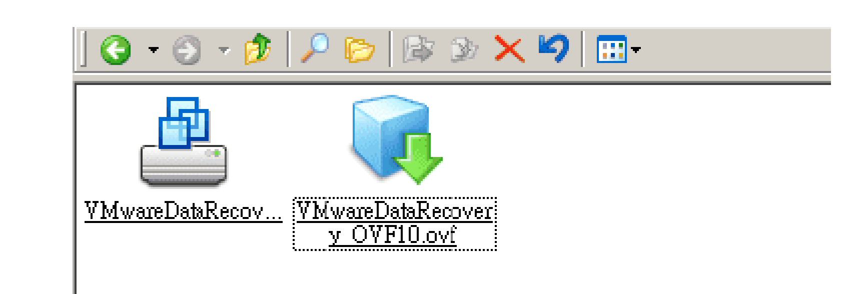
1. OVF 并不属于任何厂家

OVF 标准如其名所示，为开放、安全，可携带、高效性及可扩展的格式，主要的功能就是提

交及加壳 VM 中运行的软件服务。OVF 并不属于任何厂家或任何 Hypervisor 产品，也不属于任何处理器架构。使用 OVF 加壳好的 VA 称为 OVF 套件，其中包括了一个或多个虚拟系统，而这个虚拟系统可以在虚拟机上部署的。

2. OVF 的基本架构

一个标准的 OVF 套件包括了许多文件，并且置放在一个目录中。如果一个 OVF 套件只有一个文件的话，则称为 OVA 套件，通常是一个目录被打包成了 ZIP 格式。OVF 的主管单位称为 DMTF，在 2008 年 11 月发布了 OVF 1.0.0 之后，接着在 2010 年的 1 月也发布了 V1.1.0 版本，也是当前最新的版本。



▲ 通常有许多文件以及一个虚拟硬盘，VMware 的就是 VMDK

3. OVF 支持的厂家

大部分的虚拟机厂家都支持 OVF，但更多支持 OVF 的厂家则是软件厂家，因此我们可以看到在 vSphere 的 VA 列表中充满了各个软件厂家。当然操作系统厂家也开始支持 VA，以提供在 VM 下运行的各种环境，如大部分的 Linux 厂家都会以 OVF 的方式来发行自身新版的产品。

21.1.3 使用 Virtual Appliance

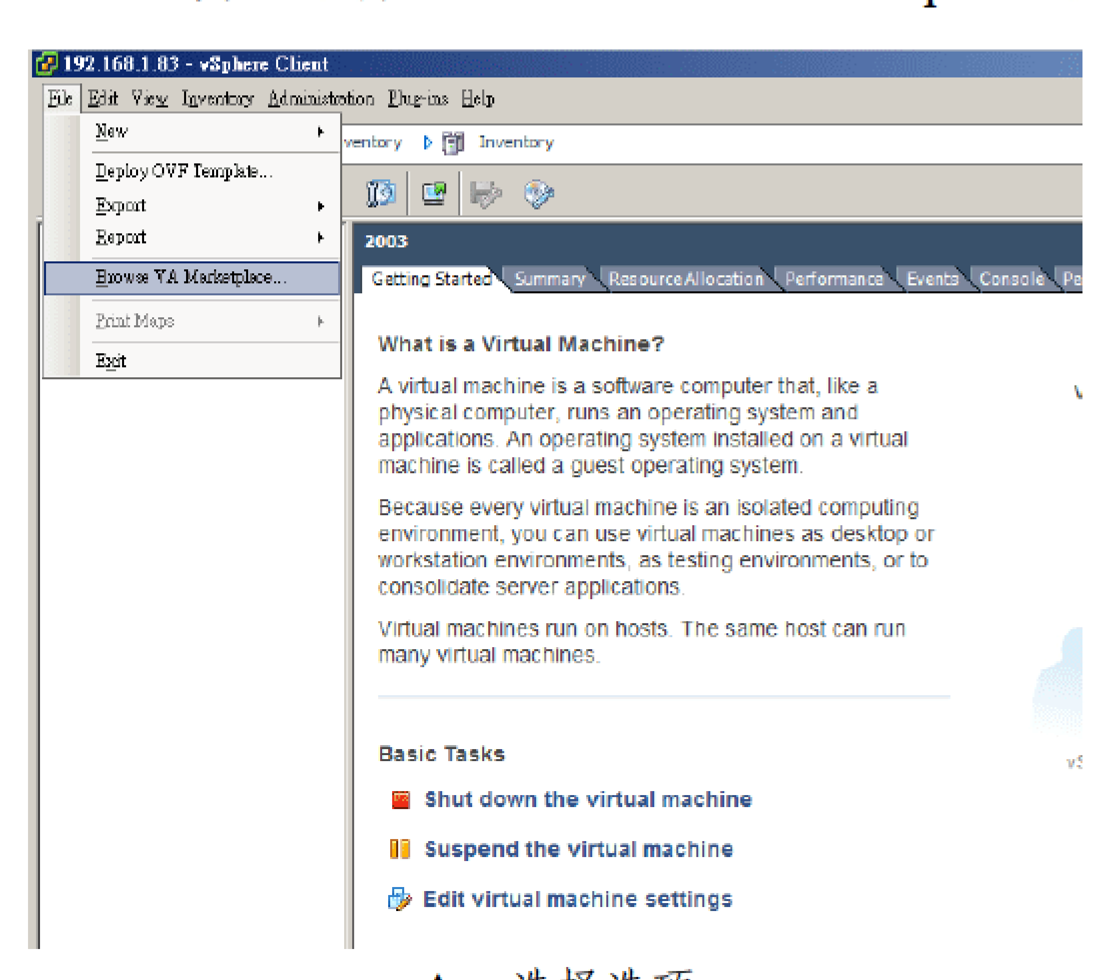
在 vSphere 下拥有一个 VA Market 功能，你可以从中选择 vSphere 推荐的 VA，而 vSphere 大部分的高级功能，如 VDR、Cisco Nexus 1000V 等服务，都是基于 VA 的方式来落实。我们就来看看在 ESX 中的 VA 功能。

1. 浏览 ESX 下的 VA

在 ESX 下有很多 vSphere 推荐的 VA，我们可以从中安装各式各样的 VA。有些 VA 是需要收费的，但大部分的 VA 是可以让你先安装，要使用 VA 提供的功能时才会开始计费。我们就来看看 vSphere 有哪些 VA。

▶ 浏览 ESX 下的 VA

1. 进入 vSphere Client，并且选择 File/Browse VA Marketplace 选项。

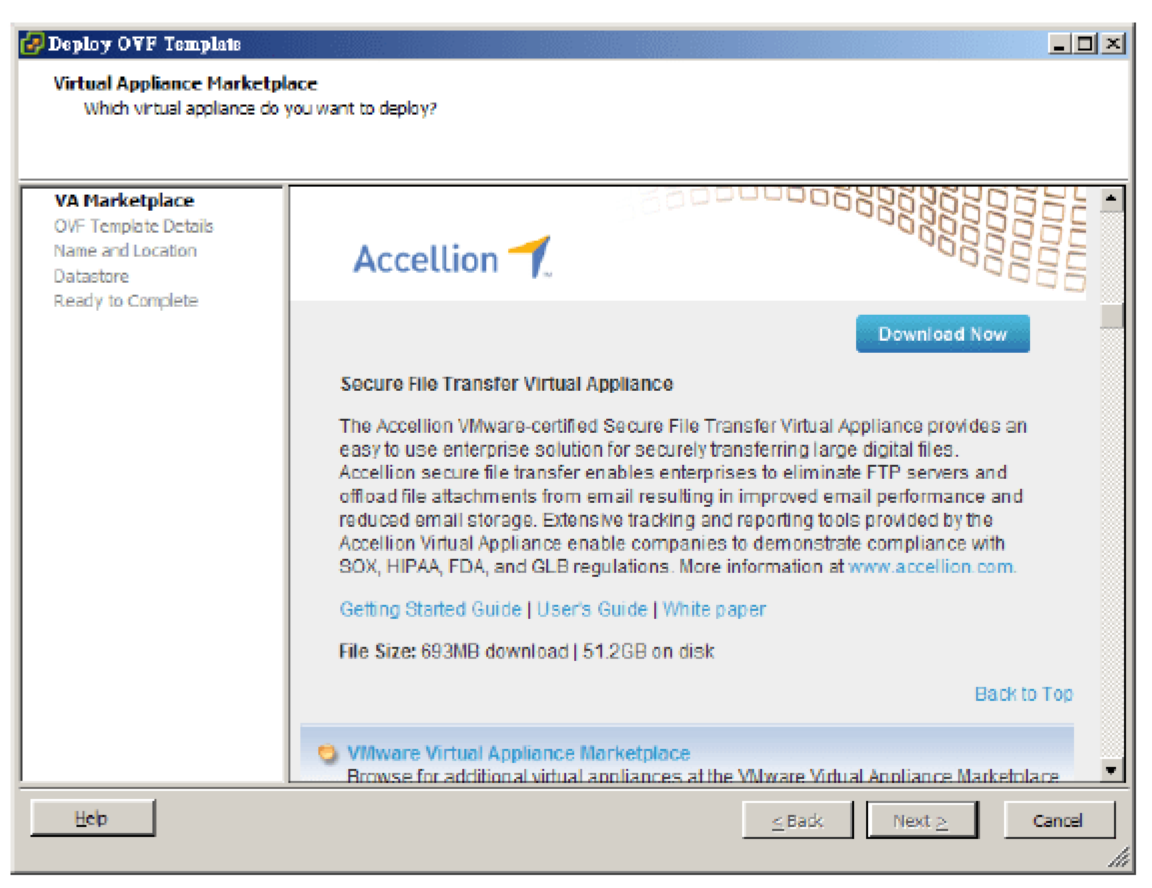


▲ 选择选项

2. 此时我们可以看到各式各样的 VA，大部分是由厂家所提供的。

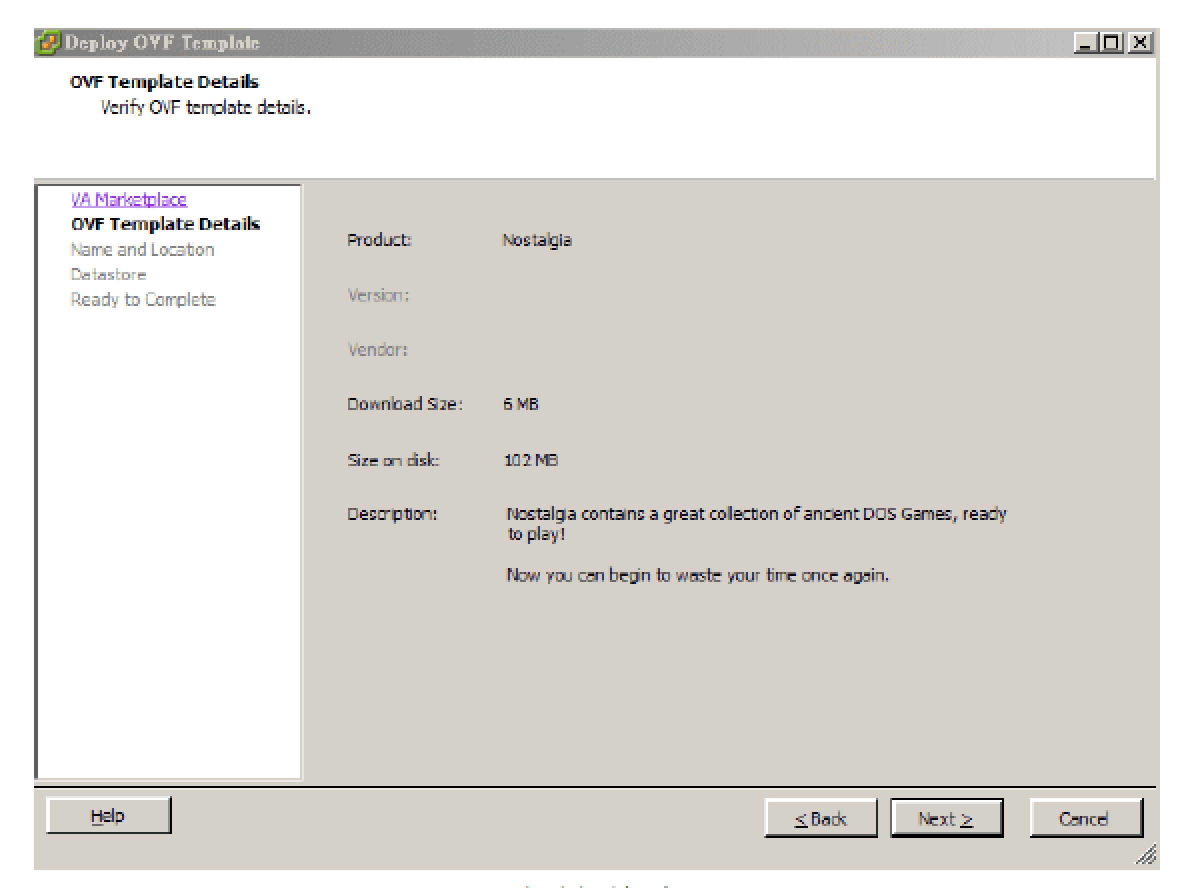
▲ 有许多 VA

3. 当你单击任何一个 VA 时，会弹出该 VA 的介绍画面。当你想要使用任何 VA 时，可以单击 Download Now 按钮。或是直接单击该 VA 的名称就可以下载。



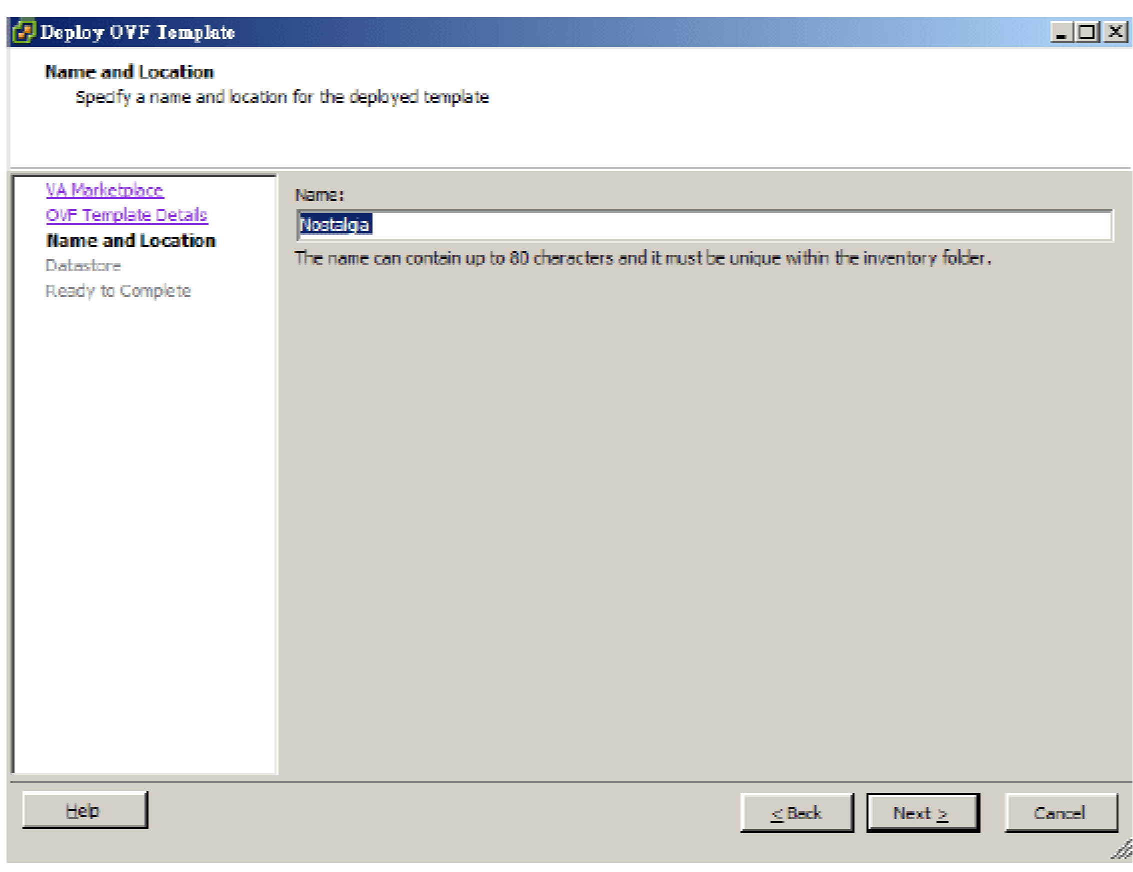
▲ 有完整介绍

4. 进入下载画面，其中会有该 OVF 的档名及大小。直接单击 Next 按钮。



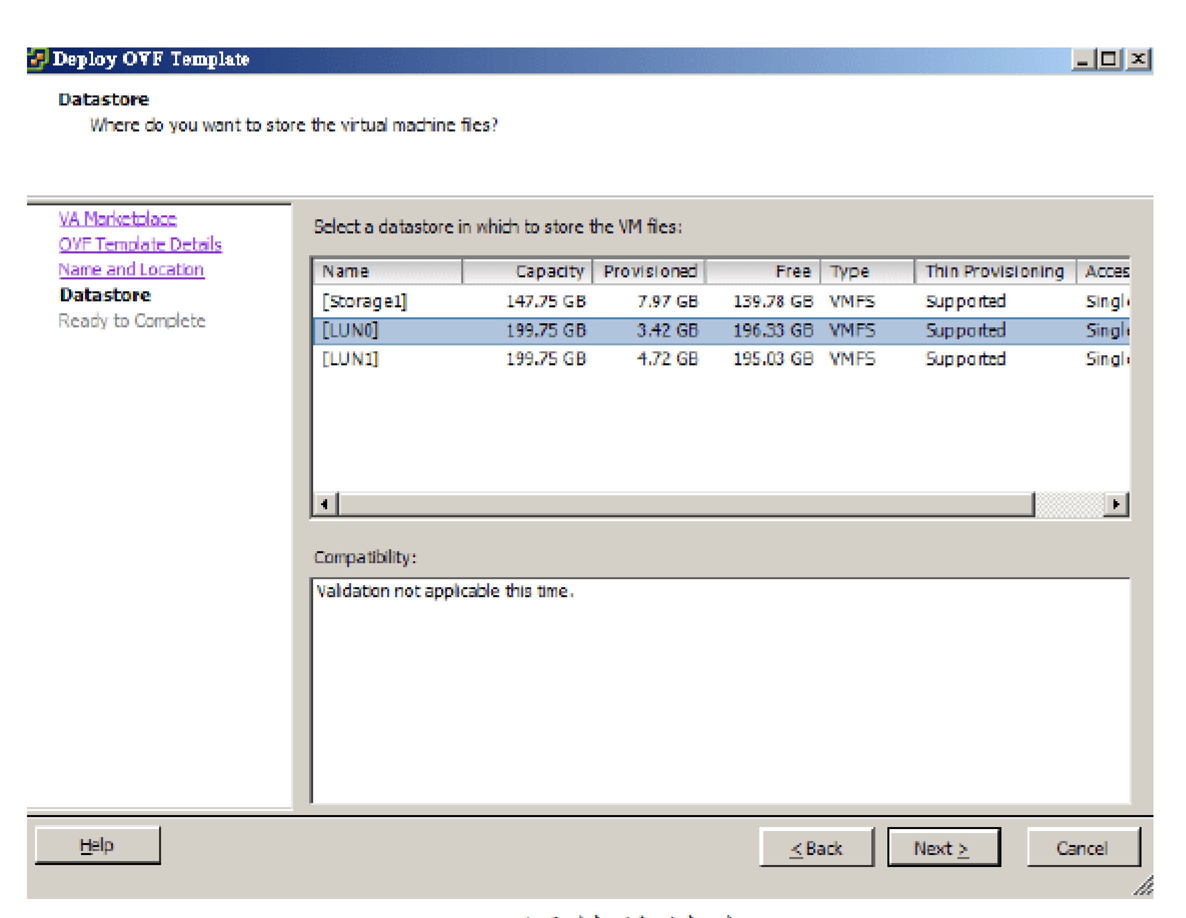
▲ 这是基本画面

5. 此时会要求你为这个 VA 的 VM 进行命名。我们就使用默认名称。这就是创建 VM 的步骤。



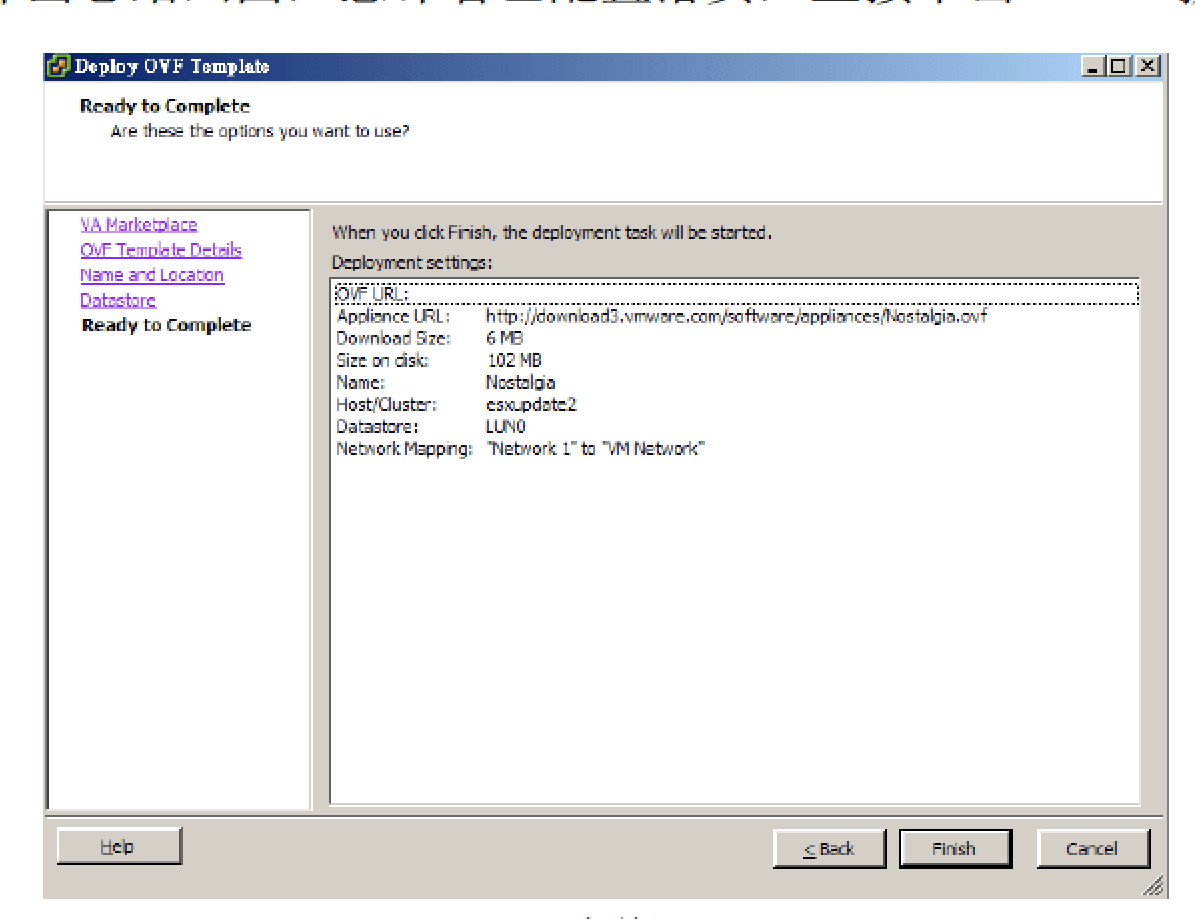
▲ 此时就是创建 VM 的步骤了

6. 接下来是选择置放的 Datastore，我们就选择如图所示选项，之后单击 Next 按钮。



▲ 置放的地方

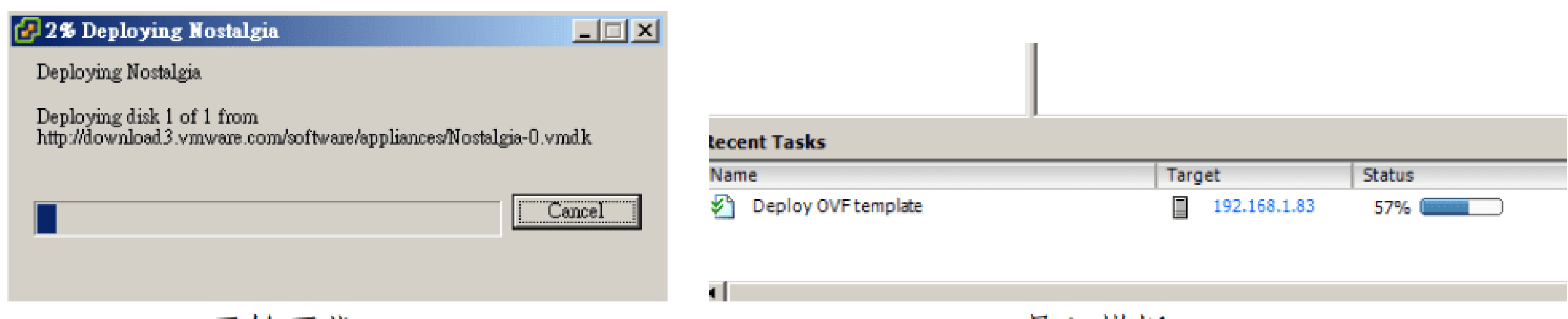
7. 接下来会弹出总结画面，意味着已配置落实，直接单击 Finish 按钮结束。



▲ 总结画面

8. 接下来将进行下载。

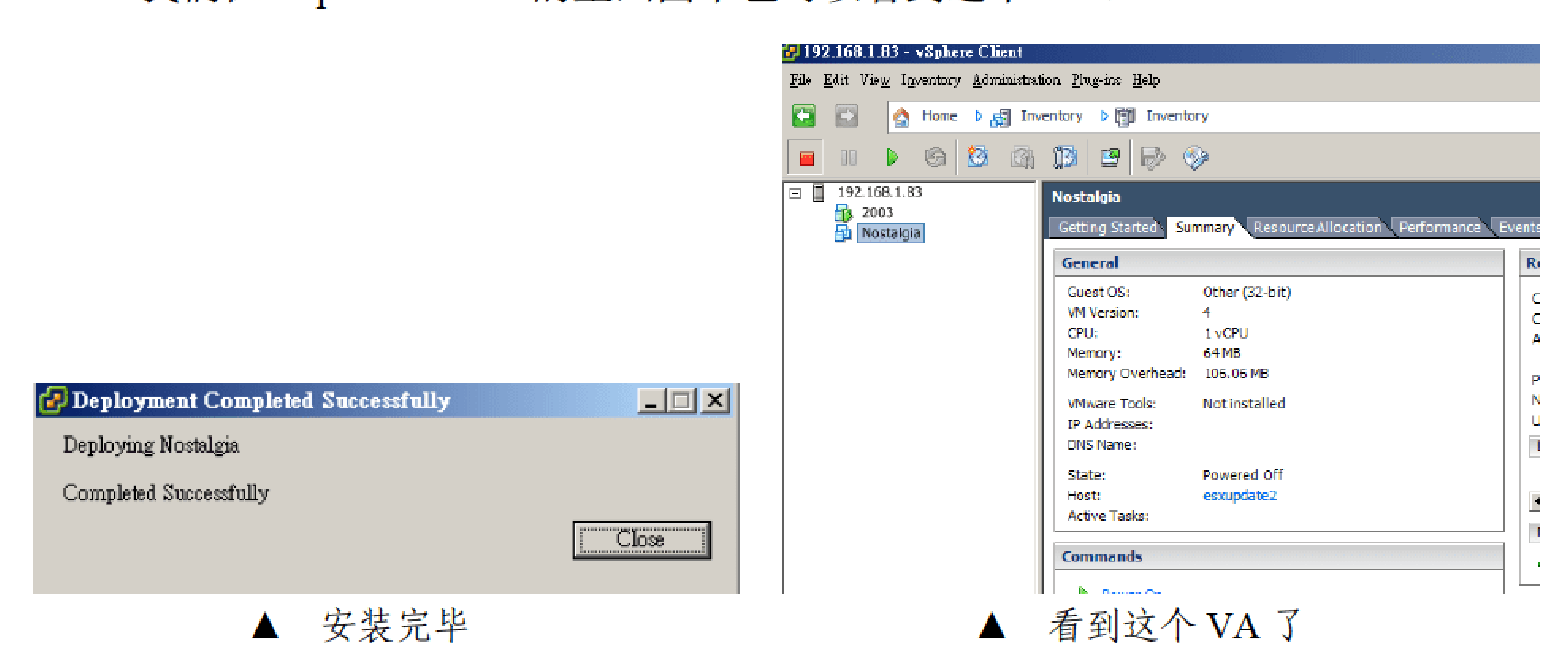
9. 在 vSphere Client 中安装 VA，也就是意味着在导入 OVF 的范本。



▲ 开始下载 VA ▲ 导入模板

10. 当弹出画面时，意味着安装完毕，单击 Close 按钮。

11. 我们在 vSphere Client 的主画面中也可以看到这个 VA。



▲ 安装完毕 ▲ 看到这个 VA 了

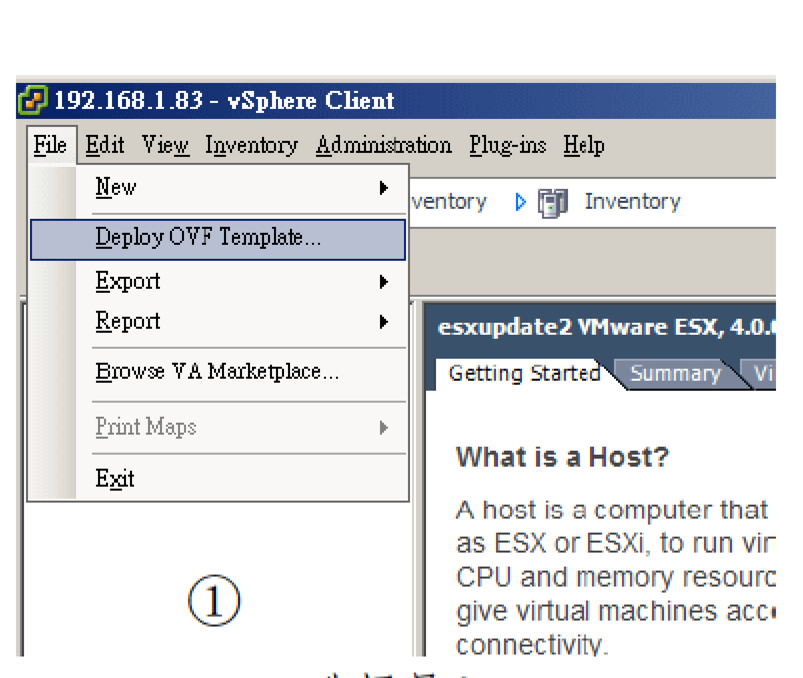
2. 简单操作 OVF 模板

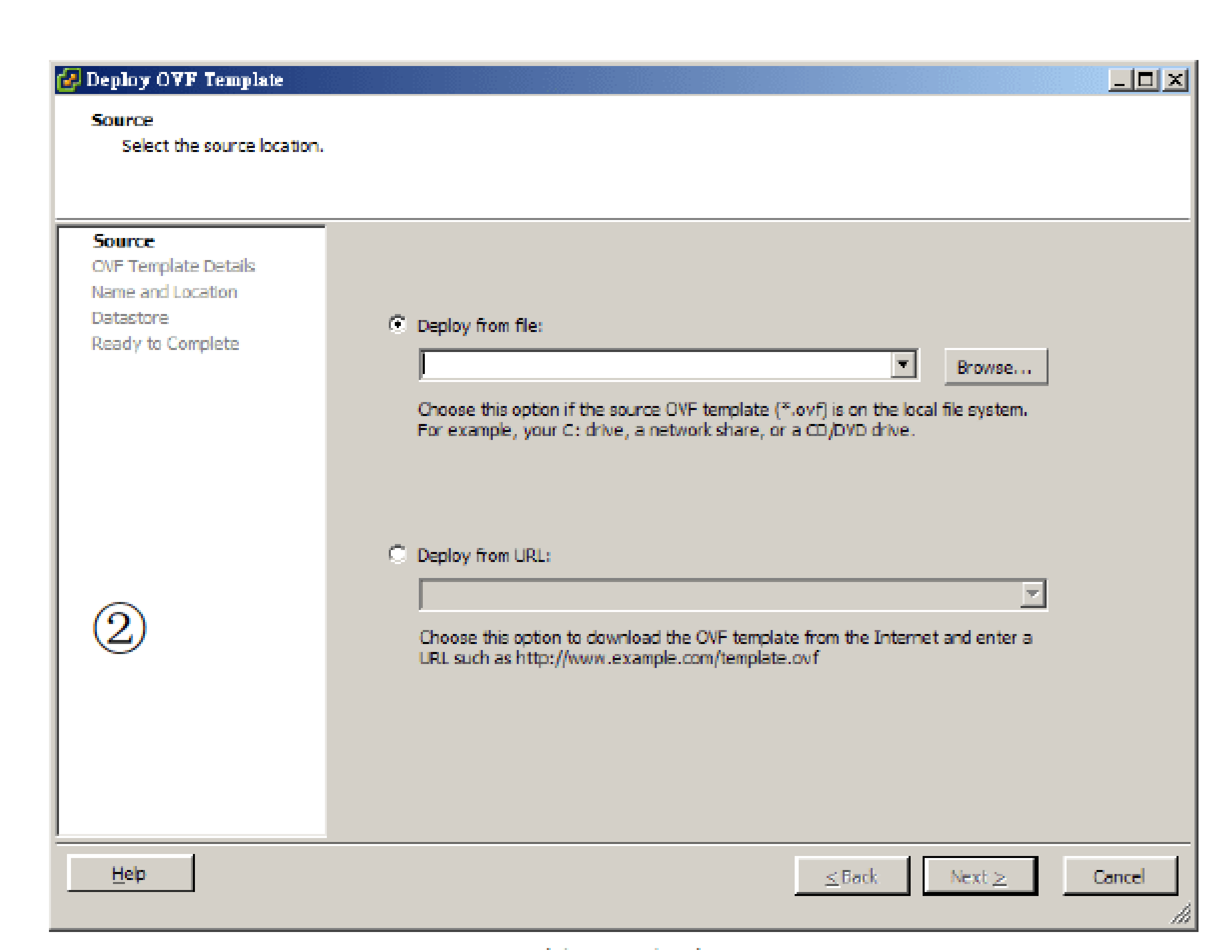
在 ESX 中操作 OVF（VA）相当简单，主要就是导出及导入等动作。

▶ 简单操作 OVF

1. 如果要导入 VA，只要在 vSphere Client 的窗口中，选择 File/Deploy OVF Template 选项。

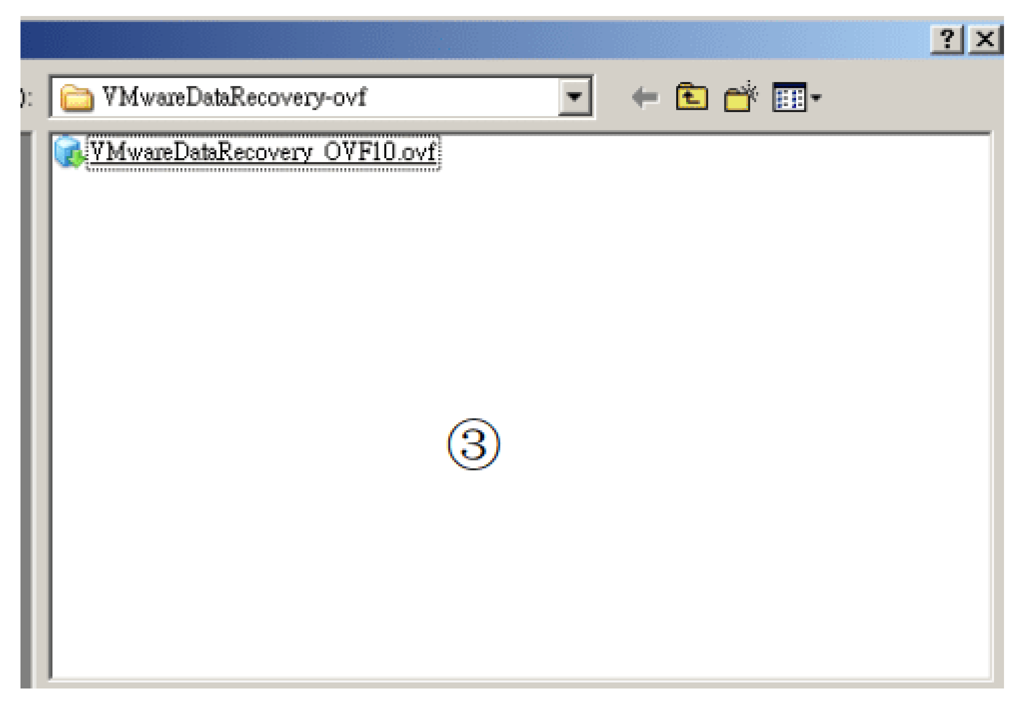
2. 此时会弹出导入的来源。



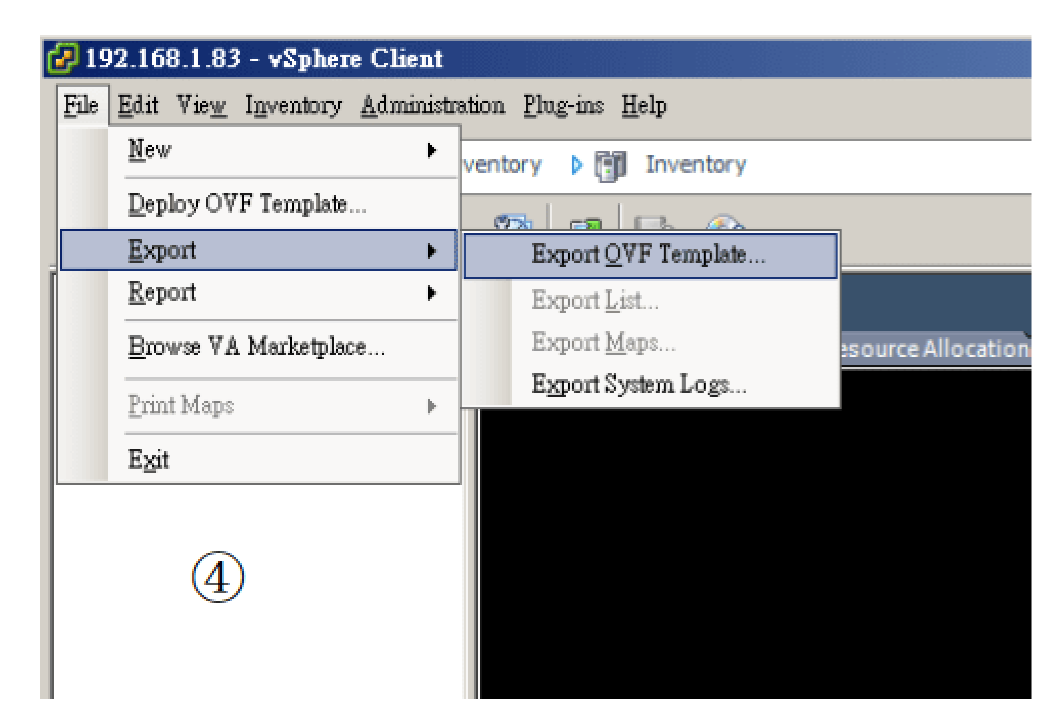


▲ 选择导入 ▲ 导入的来源

3. 单击 Browse 按钮，选择 OVF 格式的文件，就会进入之前的导入画面，只需创建名称及选择 Datastore 即可。
4. 如果要将现在的虚拟机导成 OVF，只要选择 File/Export/Export OVF Template 选项。

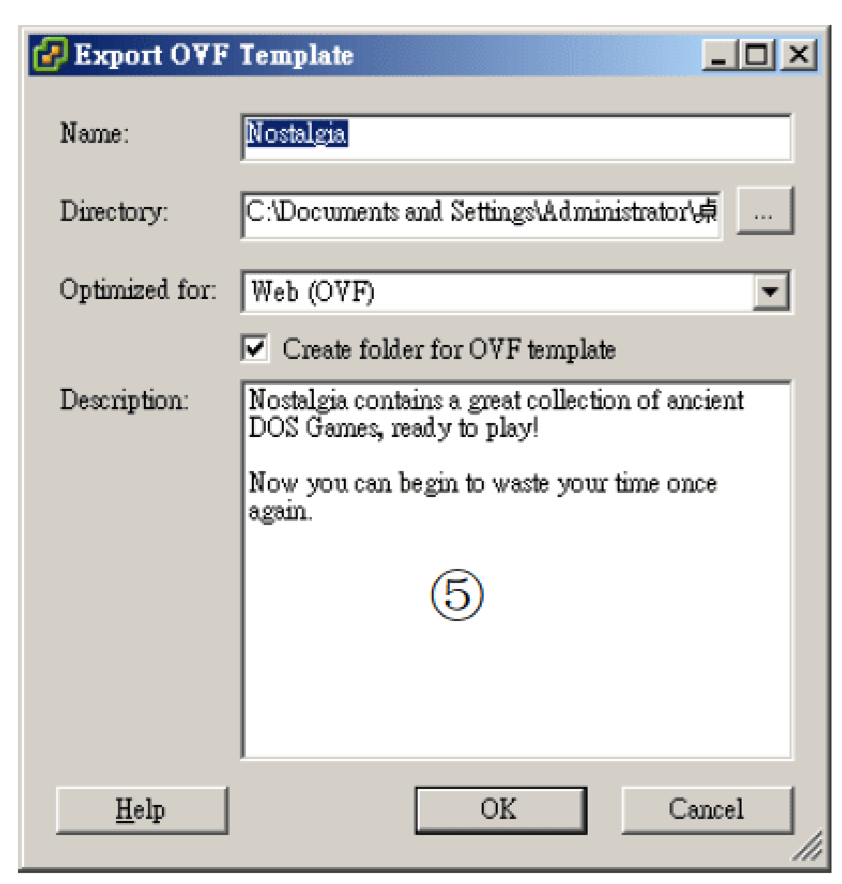


▲ 选择 OVF 文件

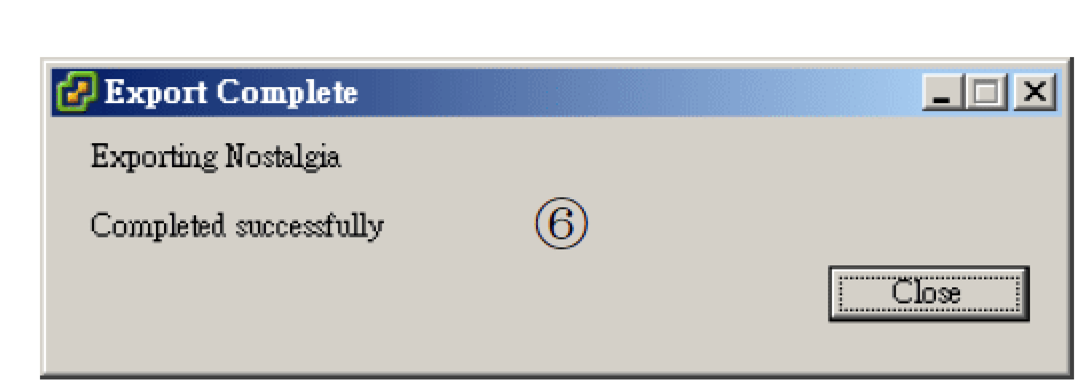


▲ 也可以将现存的 VM 导出

5. 接下来系统会要求键入名称、导出的目录、导出格式（如标准的 OVF 或是 OVA）、是否立创建文件夹等。单击 OK 按钮即开始导出。
6. 弹出的画面意味着导出完毕。

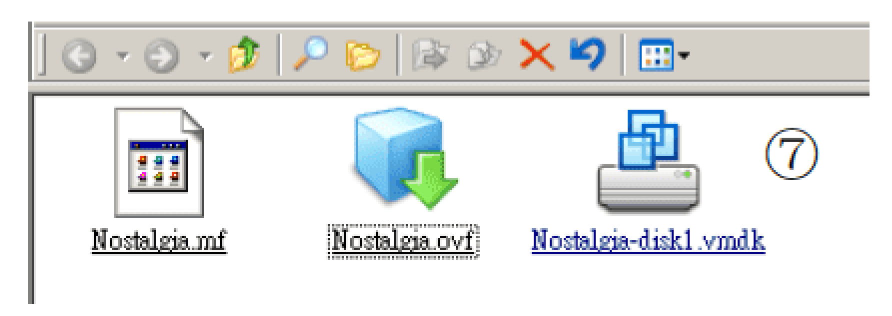


▲ 标准的导出画面



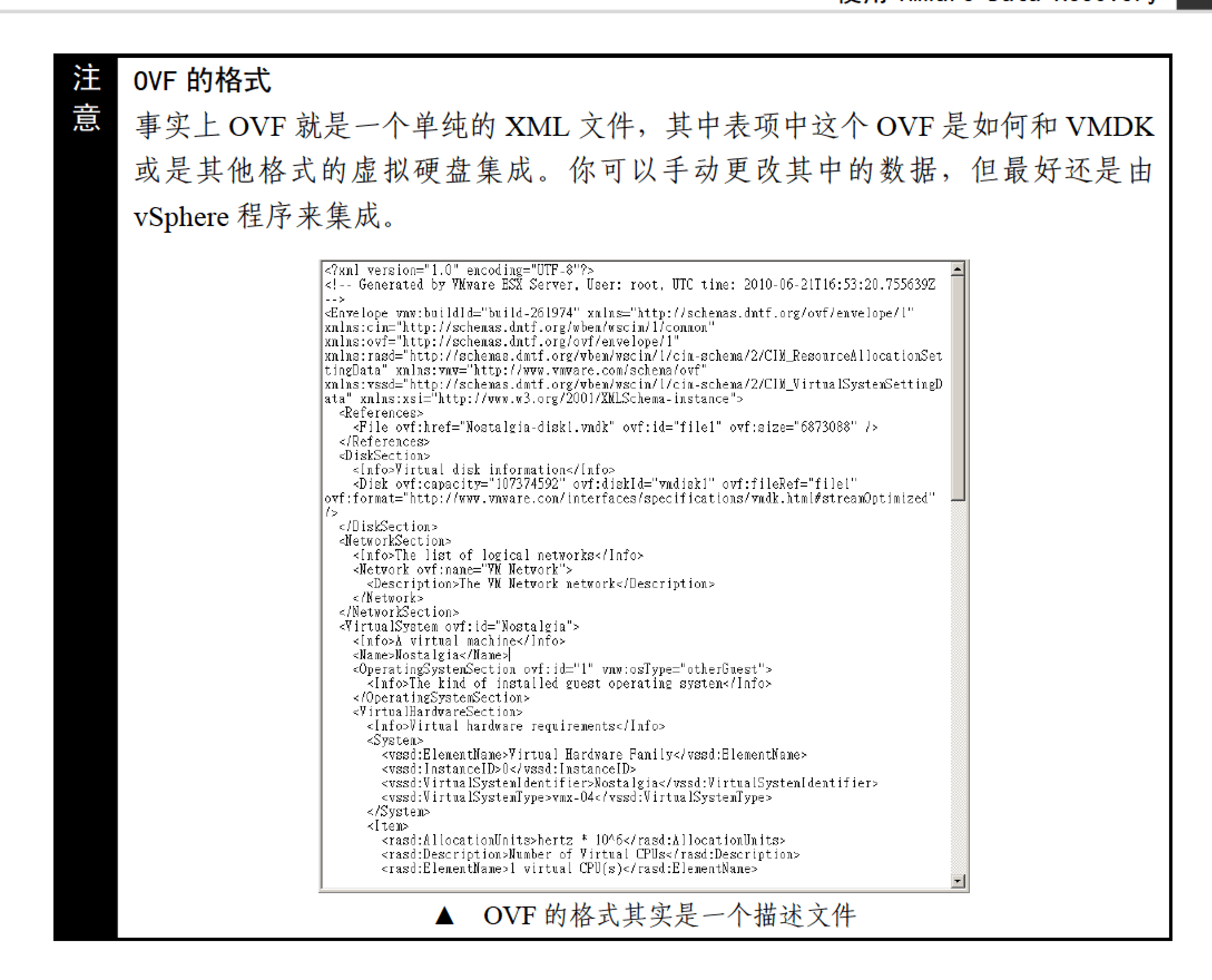
▲ 导出落实

7. 你可以在该目录下看到标准的 OVF 格式的文件。



▲ 这就是标准的 OVF 格式

> **注意**
>
> **OVF 的格式**
>
> 事实上 OVF 就是一个单纯的 XML 文件，其中表项中这个 OVF 是如何和 VMDK 或是其他格式的虚拟硬盘集成。你可以手动更改其中的数据，但最好还是由 vSphere 程序来集成。




```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Generated by VMware ESX Server, User: root, UTC time: 2010-06-21T16:53:20.755639Z
-->
<Envelope vmw:buildId="build-261974" xmlns="http://schemas.dmtf.org/ovf/envelope/1"
xmlns:cim="http://schemas.dmtf.org/wbem/wscim/1/common"
xmlns:ovf="http://schemas.dmtf.org/ovf/envelope/1"
xmlns:rasd="http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_ResourceAllocationSet
tingData" xmlns:vmw="http://www.vmware.com/schema/ovf"
xmlns:vssd="http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_VirtualSystemSettingD
ata" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <References>
    <File ovf:href="Nostalgia-disk1.vmdk" ovf:id="file1" ovf:size="6873088" />
  </References>
  <DiskSection>
    <Info>Virtual disk information</Info>
    <Disk ovf:capacity="107374592" ovf:diskId="vmdisk1" ovf:fileRef="file1"
ovf:format="http://www.vmware.com/interfaces/specifications/vmdk.html#streamOptimized"
/>
  </DiskSection>
  <NetworkSection>
    <Info>The list of logical networks</Info>
    <Network ovf:name="VM Network">
      <Description>The VM Network network</Description>
    </Network>
  </NetworkSection>
  <VirtualSystem ovf:id="Nostalgia">
    <Info>A virtual machine</Info>
    <Name>Nostalgia</Name>
    <OperatingSystemSection ovf:id="1" vmw:osType="otherGuest">
      <Info>The kind of installed guest operating system</Info>
    </OperatingSystemSection>
    <VirtualHardwareSection>
      <Info>Virtual hardware requirements</Info>
      <System>
        <vssd:ElementName>Virtual Hardware Family</vssd:ElementName>
        <vssd:InstanceID>0</vssd:InstanceID>
        <vssd:VirtualSystemIdentifier>Nostalgia</vssd:VirtualSystemIdentifier>
        <vssd:VirtualSystemType>vmx-04</vssd:VirtualSystemType>
      </System>
      <Item>
        <rasd:AllocationUnits>hertz * 10^6</rasd:AllocationUnits>
        <rasd:Description>Number of Virtual CPUs</rasd:Description>
        <rasd:ElementName>1 virtual CPU(s)</rasd:ElementName>
```


▲ OVF 的格式其实是一个描述文件

21.2 使用 VMware Data Recovery 备份及还原

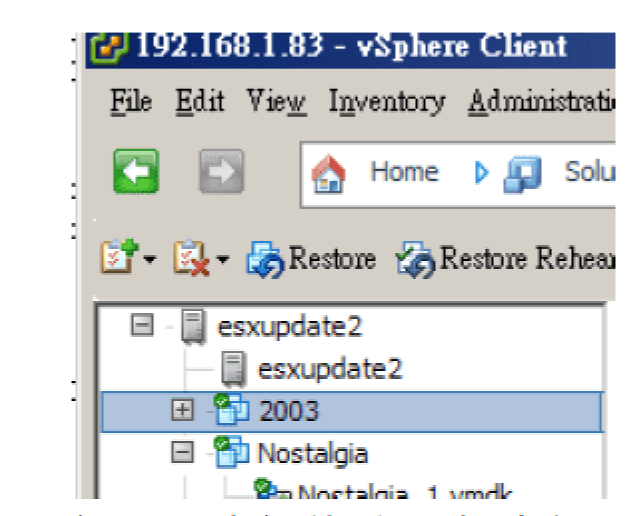
在 vSphere 之后，VMware 的备份还原产品热点不再是 VCB。除了入门两个版本的 vSphere 之外，其他所有版本的 vSphere 都开始使用全新的 VMware Data Recovery（简称 VDR）作为备份还原的工具，而 VDR 正是 VMware VA 的一个最好示例，我们在这一节就来看看如何使用 VDR 来备份及还原 VM。

21.2.1 VDR 的特色

VDR 使用和 VCB 是一样的原理，将运行中的 VM 做出快照，并且将快照备份起来，这样一来可以让整个 VM 继续运行，并发也可以进行备份。但和 VCB 比起来，VDR 的优点强上太多了，下面就来看看 VDR 的特色。

1. 使用“保护”VM 的概念

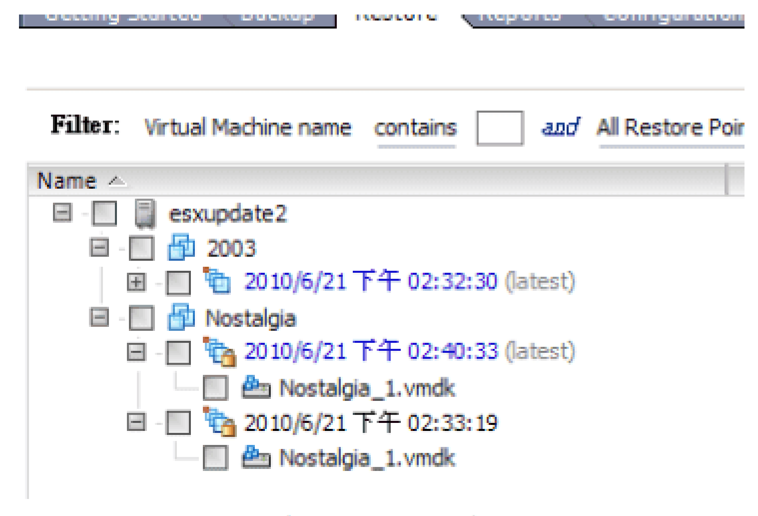
在 VDR 中，VM 是受“保护”的。当一个 VM 被 VDR 保护之后，意味着这个 VM 已经被 VDR 完整备份了，接下来就可以备份补丁过的数据，因此在 VDR 中，我们会以“保护”的观念来操作 VM 及 VDR。



▲ 当一个 VM 被保护时，会弹出不同的图示

2. 可以备份完整的 VM 和增量备份

当一个 VM 被 VDR 保护之后，意味着这个 VM 已经被 VDR 备份过完整的数据了。当这个 VM 再有变动之后，VDR 可以只备份新建的数据，称为“增量备份”（Incremental Backup）。这些新建数据可以精确到文件或目录等级，这和主流的备份软件概念完全一样，用户可以充分利用 VDR 的多种备份机制操作。



▲ 会有多个还原点可以选择

3. 完全不需要安装 Agent

大部分支持完整及增量备份的产品，都需要在主机中安装备份代理程序（Agent）。但是受 VDR 保护的 VM 完全不需要安装 Agent，只要在 vCenter 中配置 VDR 被保护，就可以充分享受 VDR 的高级功能。

4. 充分利用 vCenter Server 的调度机制

VDR 可充分利用 vCenter Server 的调度机制来备份 VM。只要将 VDR 安装在 vCenter 中，你就可以定时定量来备份 VM，当然你也可以将 VDR 安装在 vSphere Client 中，但功能就没有这么强大了。

5. 数据的克隆删除

在较高级的存储设备中才有重复删除（De-duplicate）的功能，但是 VDR 本身就是一个带有 De-duplicate 功能的 VA，因此可以省下大量的存储空间，更棒的是可以加快备份和还原的过程。

21.2.2 VDR 备份和还原的过程

在开始使用 VDR 之前，我们必须先理解 VDR 备份和还原 VM 的过程。

1. 参加 VDR 的角色

完整的 VDR 必须有三个部件，我们就来看看。

▶ 参加 VDR 的三个部件

- 安装到 vSphere Clietn 或是 vSphere Server 的 VDR Plug-in。
- VMware 提供的 VDR VA（在 VDR 的光盘片中），使用 OVF。
- 有 De-duplicated 功能的独立存储。

2. VDR 备份 VM 的过程

备份和还原都是通过一个专门的 VMware VA 处理。备份的原理也是创建快照之后再进行备份，以免 VM 的服务中断，我们就来看看 VDR 备份 VM 的过程。

▶ VDR 备份 VM 的过程

- 使用 vCenter 进行 VM 的选择、备份的调度、数据的保留、策略的决定以及备份目的的选择。
- 落实之后，将选择的结果指派到 VDR 这个 VA 中。
- VA 接到指派任务后，开始进行该 VM 的快照制作。
- 开始进行备份任务，确定至少在快照创建的那一时间点，数据都会被备份起来。
- VDR 将快照挂载到自身的 VM 中，以保证网络带宽不会被备份占据。当快照被挂载之后，数据就会直接压入网络存储设备。
- 在备份过程 VDR 会遍历是否有重复数据，并且删除重复的数据。在数据确定干净之后，才会真正可写备份的目的地。
- 当数据备份落实后，VDR 会将快照卸除，并且将之前的快照给卸除。

3. 使用 VDR 还原 VM

备份的目的就是在数据丢失时还原，我们就来看看 VDR 还原 VM 的步骤。

▶ VDR 还原 VM 的步骤

- 在还原完整的 VM 时，VDR 会呈现每一个 VM 选择还原点及时间的那一个点。
- 当用户给定了还原时间点，就会开始还原 VM 至给定的 ESX 或是资源池上。
- 当还原至现存的 VM 上，VDR 只会还原改动过的数据。
- 当然也可以还原文件，但还原的过程需要 VM 本身内部的操作系统来落实。

21.3 安装及配置 VDR

要让 vCenter 具有 VDR 功能，主要有几个步骤，而安装了 VDR 之后，vCenter 会自动集成一个 Plug-in 以及会新建一个 VM，该 VM 就是运行 VDR 的 VA。

21.3.1 安装 VDR

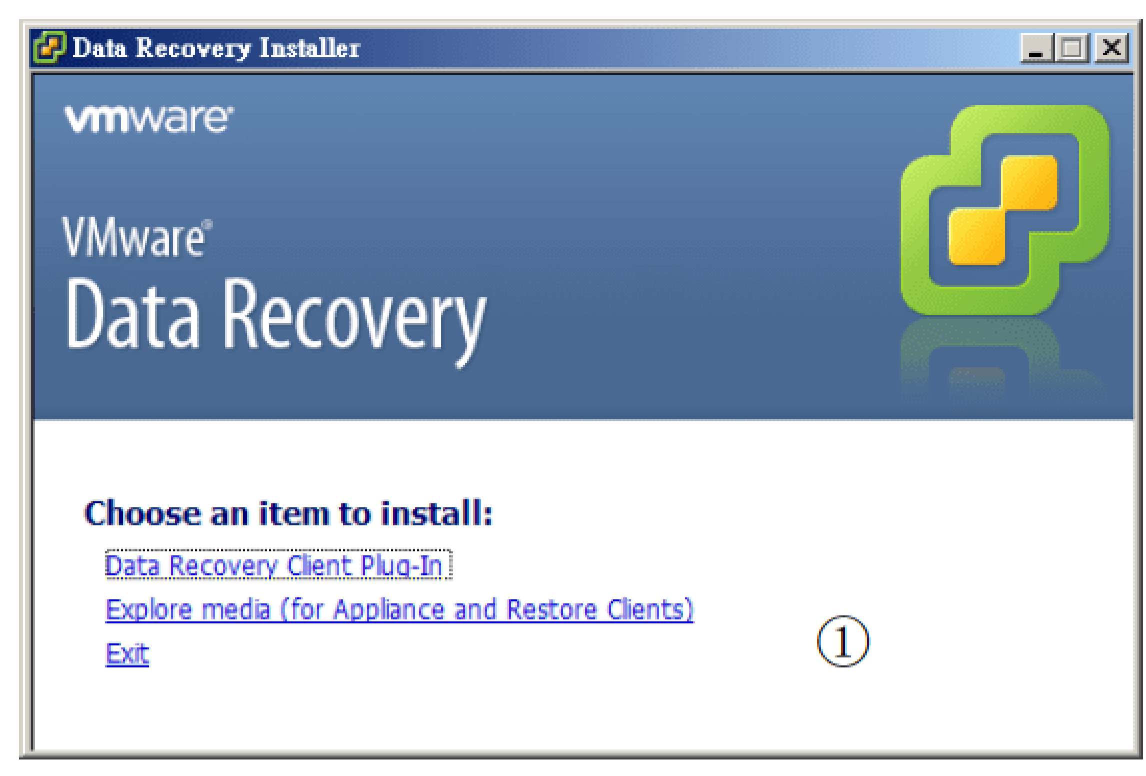
安装 VDR 分为两个步骤，第一个为安装 vCenter/vSphere Client 的 Plug-in，第二则是安装 VDR 的 VA，下面就是详细步骤。

1. 安装 vCenter 的 Plug-in

安装 VDR 中 vCenter 的 Plug-in，必须在拥有 vSphere Client 的机器上进行，才会安装到 vCenter 或 ESX 中成为 Plug-in。下面就是详细的安装步骤。

▶ 安装 VDR 的 Plug-in 到 vCenter 中

1. 插入 VDR 的光盘，或是挂入 ISO 档到虚拟光盘上，此时会弹出 VDR 的安装画面，选择 Data Recovery Client Plug-In。
2. 接下来会进入安装画面，单击 Next 按钮继续。



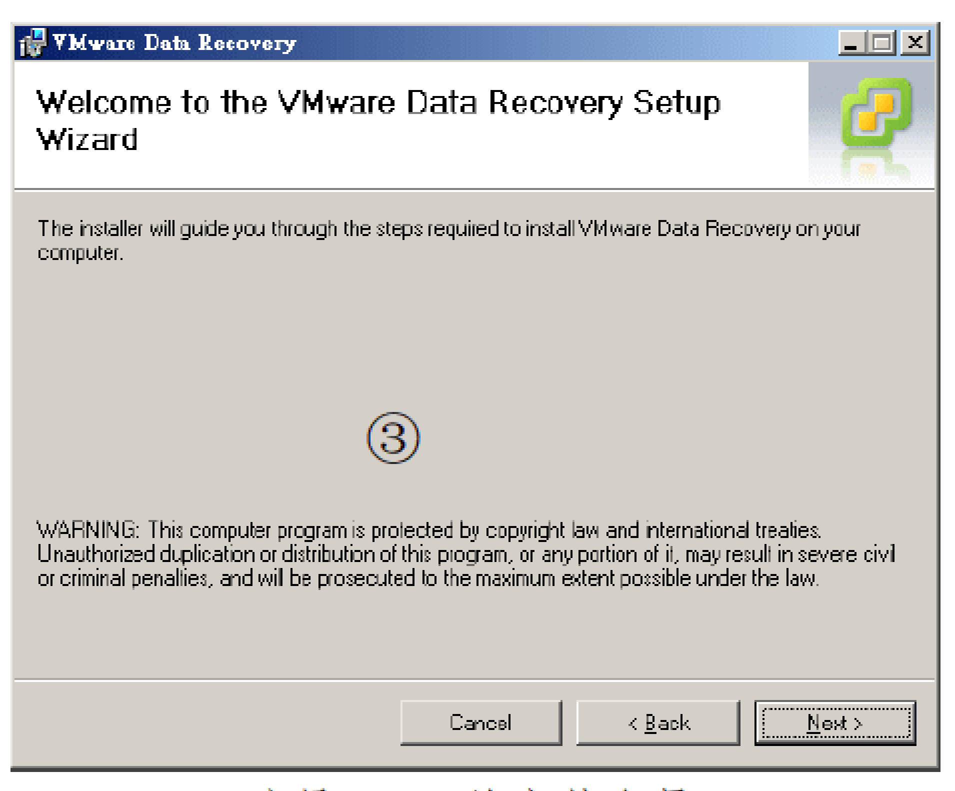
▲ 选择安装



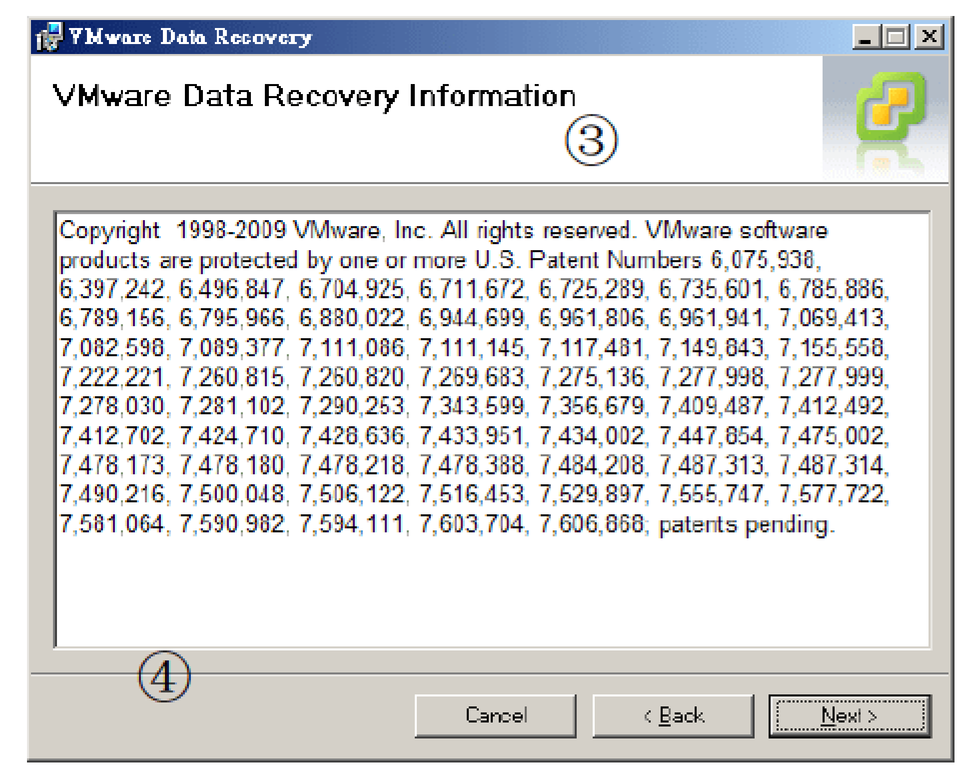
▲ 进入安装画面

3. 进入 VDR 配置向导，单击 Next 按钮继续。

4. 接下来会显示版本编号，在截稿为止，VDR 最新的版本为 1.2，单击 Next 按钮继续。



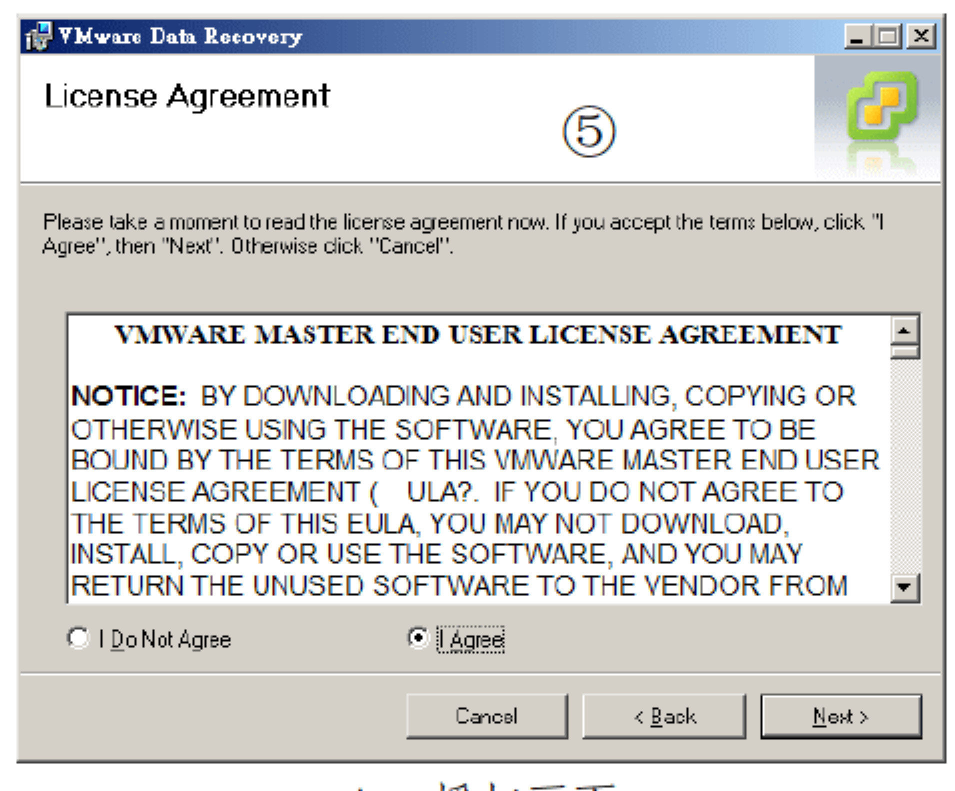
▲ 这是 VDR 的安装向导



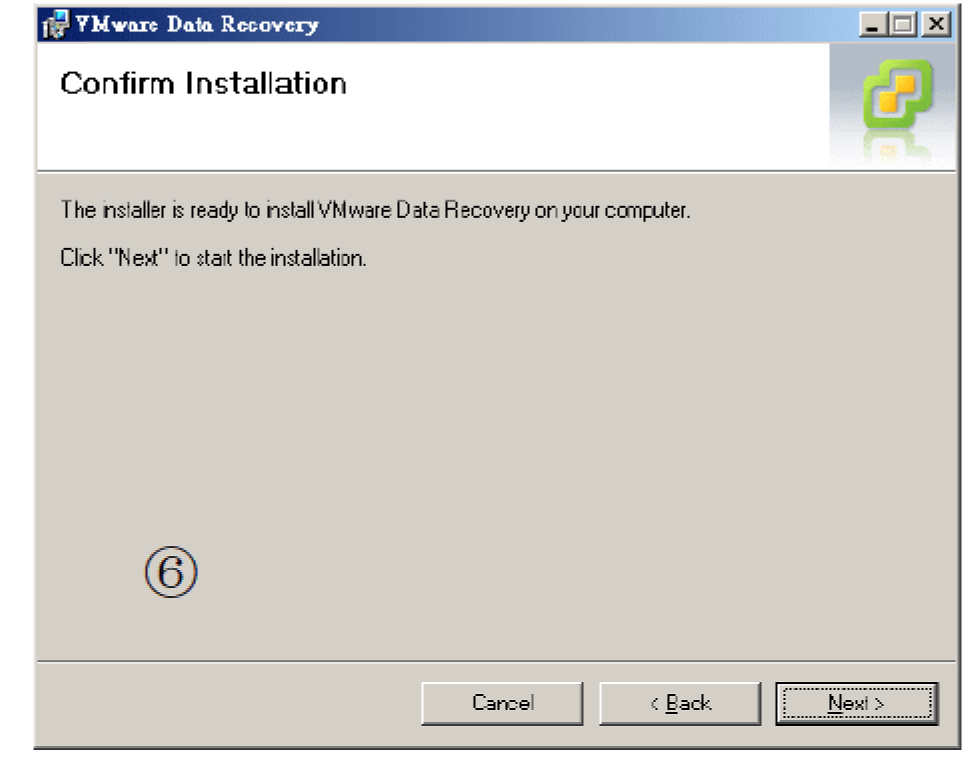
▲ 当前最新的版本是 1.2，本章演示的是 1.1

5. 接下来是授权，单击 I Agree 单选按钮之后，单击 Next 按钮继续。

6. 接下来会进行物理安装，单击 Next 按钮继续。



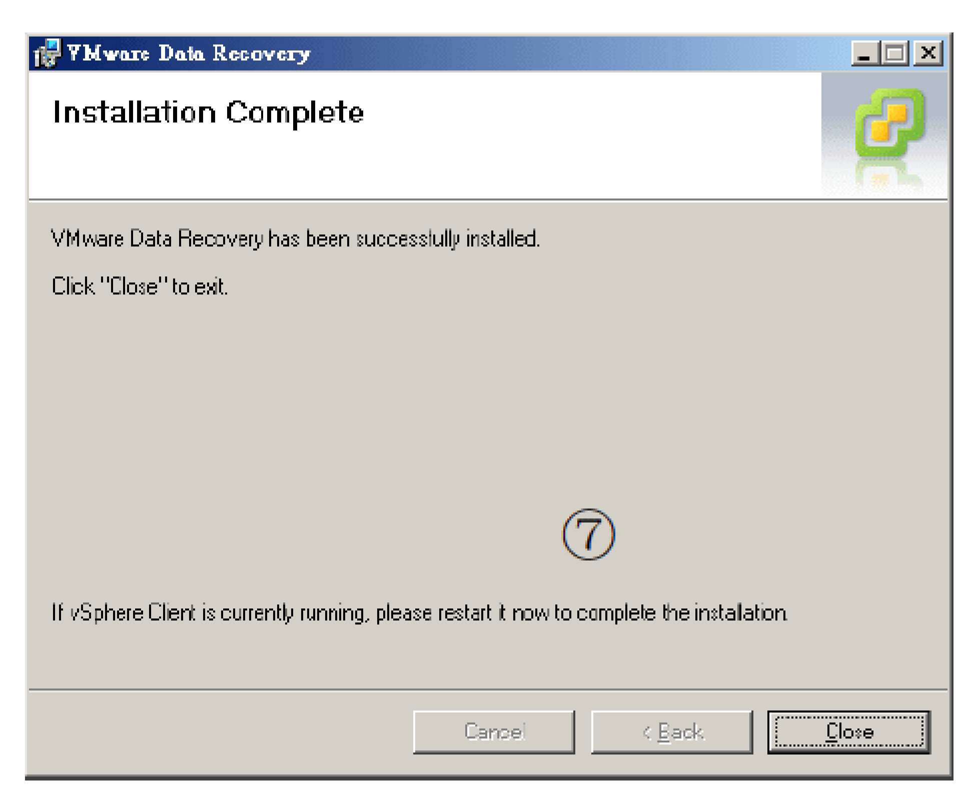
▲ 授权画面



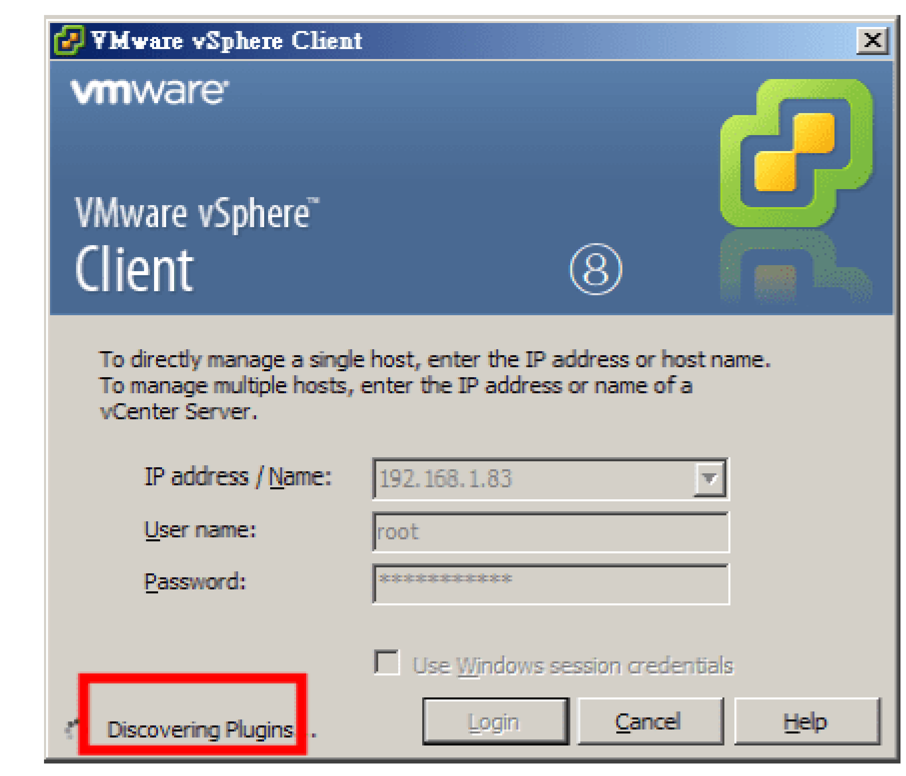
▲ 开始安装

7. 当弹出如下图所示的画面时，意味着安装完毕，单击 Close 按钮。

8. 接下来可以进入 vCenter Server 或是 vSphere Client，在激活时，可以看到 Plug-in 的装载。

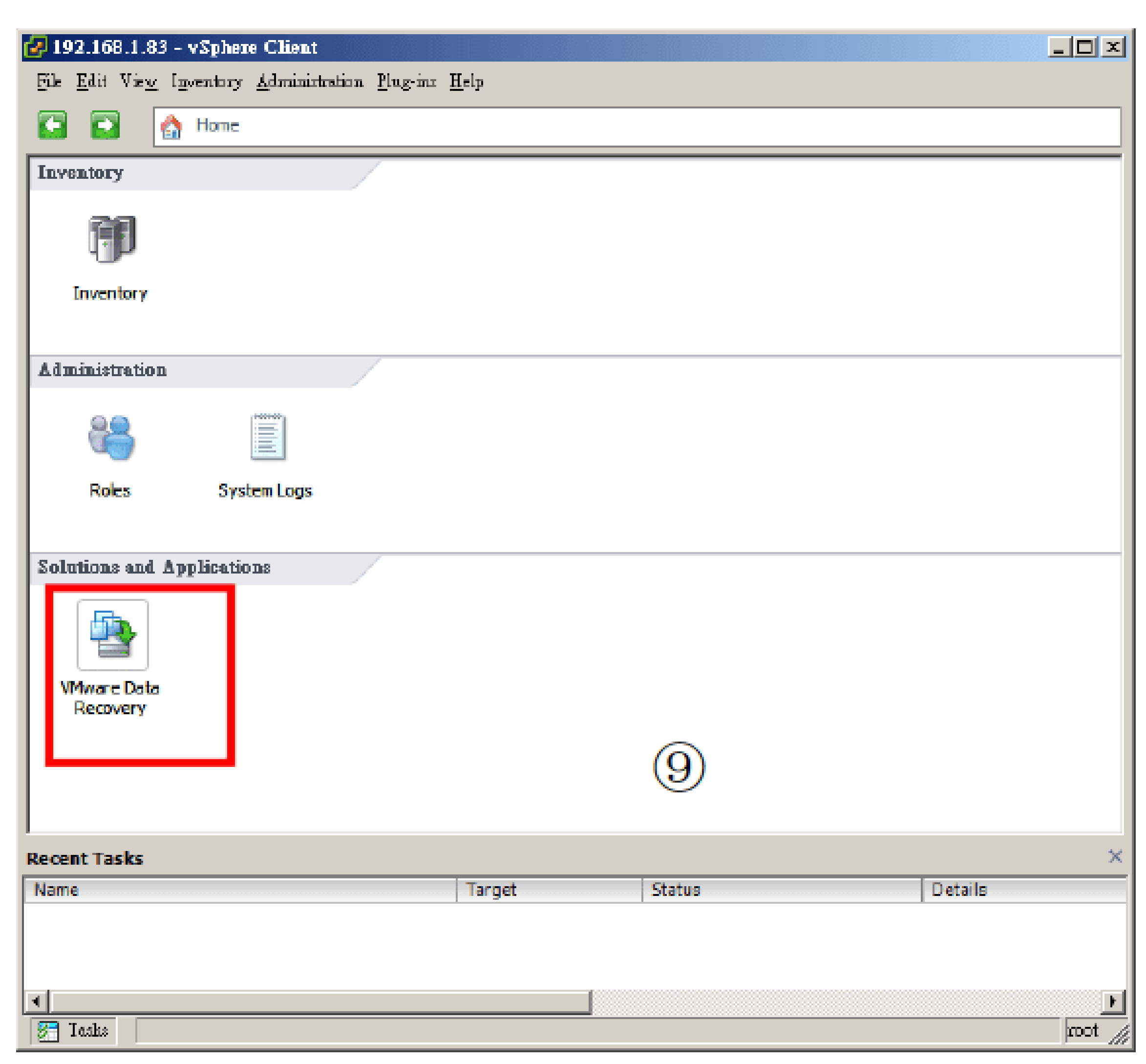


▲ 安装完毕



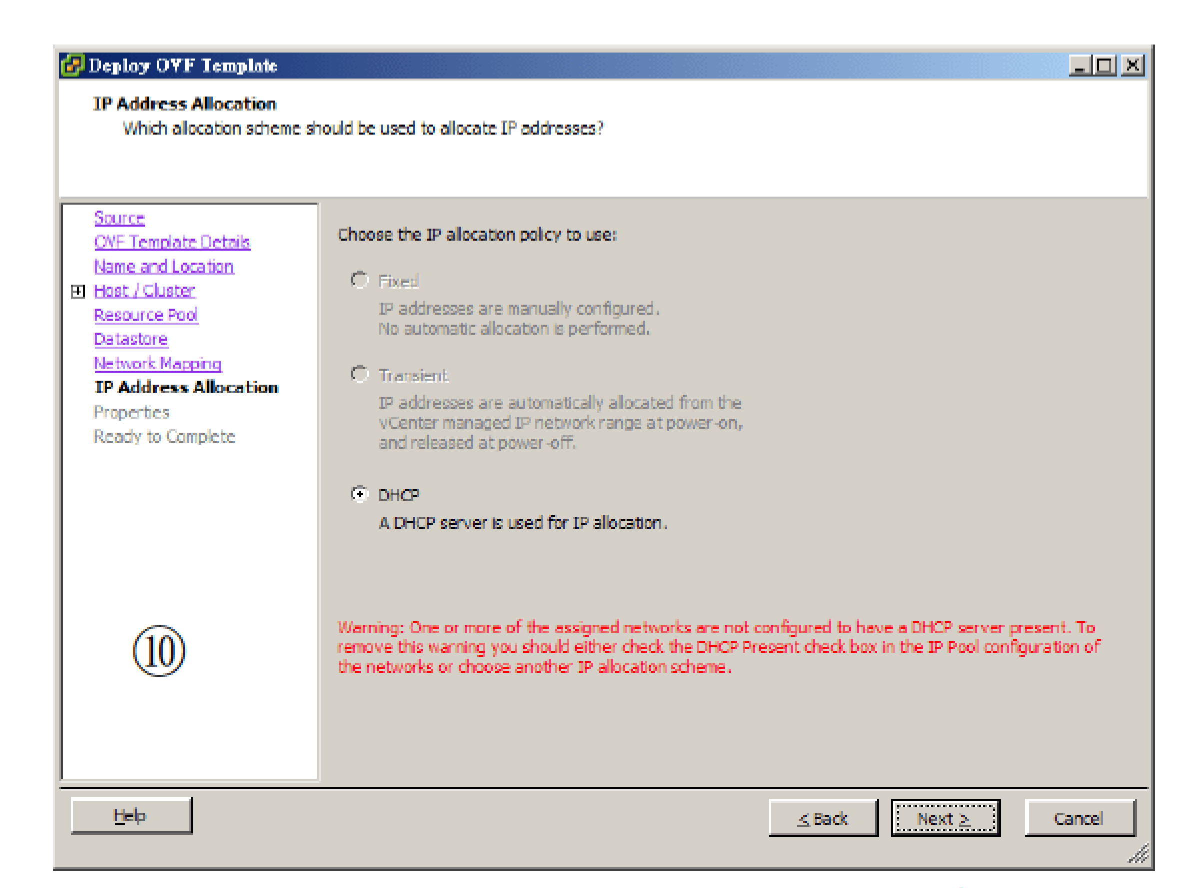
▲ 正在装载 Plug-in

9. 在进入 vSphere Client 之后，会看到在“Home”选项卡下，已经有一个安装好的 VA 了，就是 VDR。



▲ VDR 已经安装成功了

10. 在 vCenter 中也可以看到 VDR 也成功安装了。



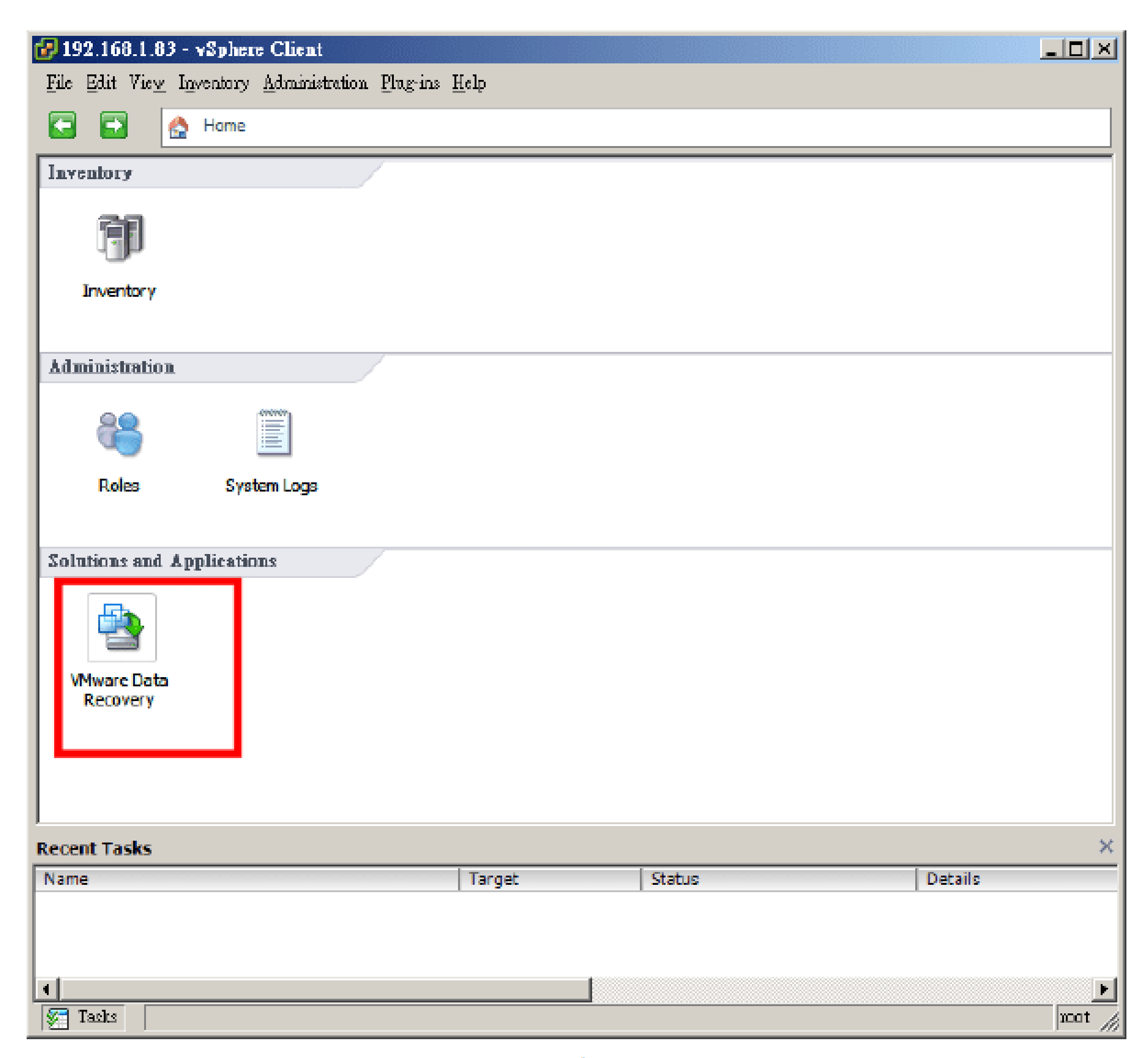
▲ 在 vCenter 中导入 VDR 的选项较多

2. 在 vSphere Client 中配置 VDR

当我们安装好 VDR 在 vSphere Client 的 Plug-in 之后，就要开始配置这个 VA 了。由于 VDR 是一个 VA，也是在 VM 中运行，因此接下来就是要将这个 VA 安装到 vSphere Client 中。下面就是步骤。

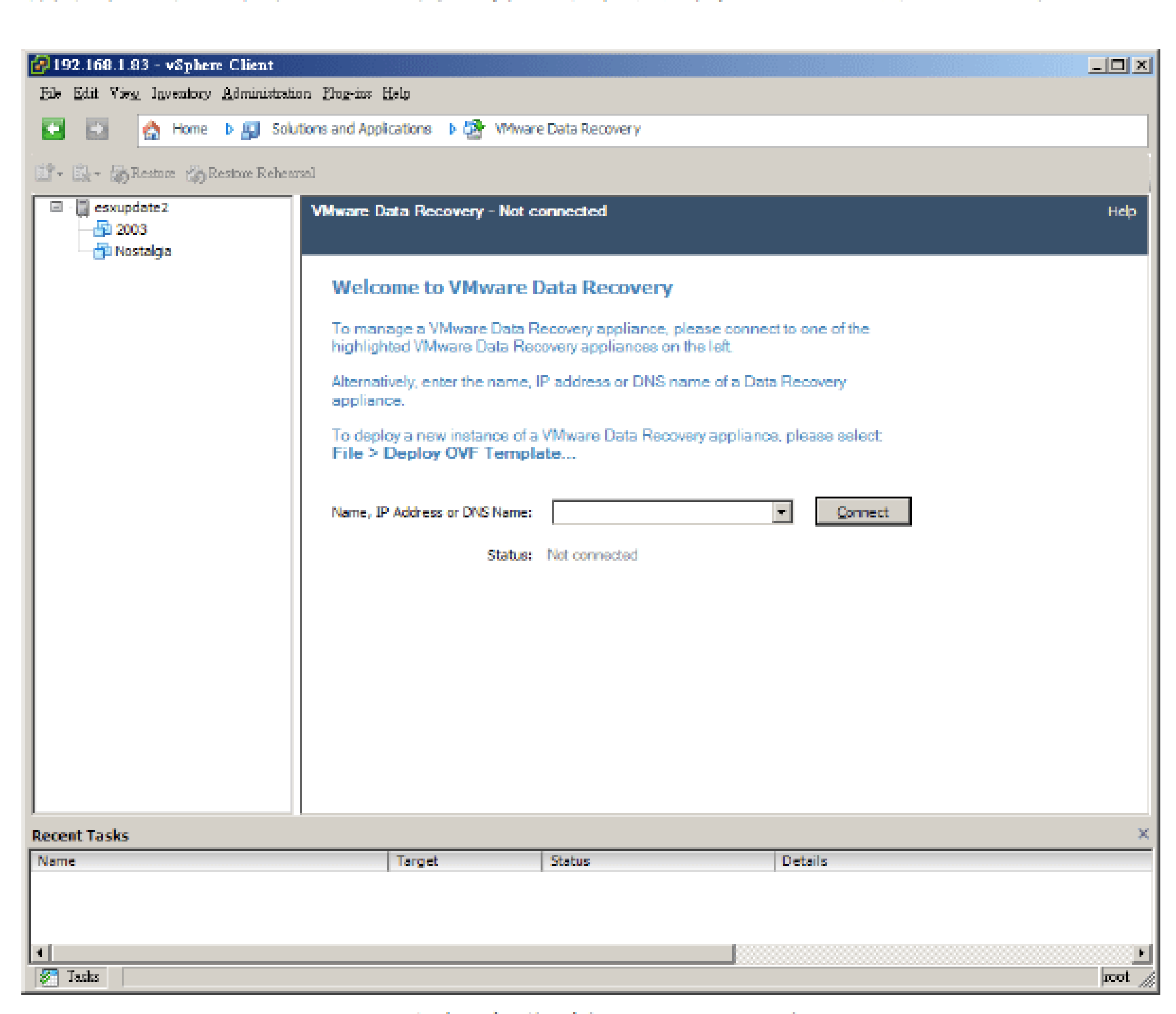
▶ 将 VDR 的 VA 安装到 vSphere Client 中

1. 首先我们进入 vSphere Client，并且运行 VDR。



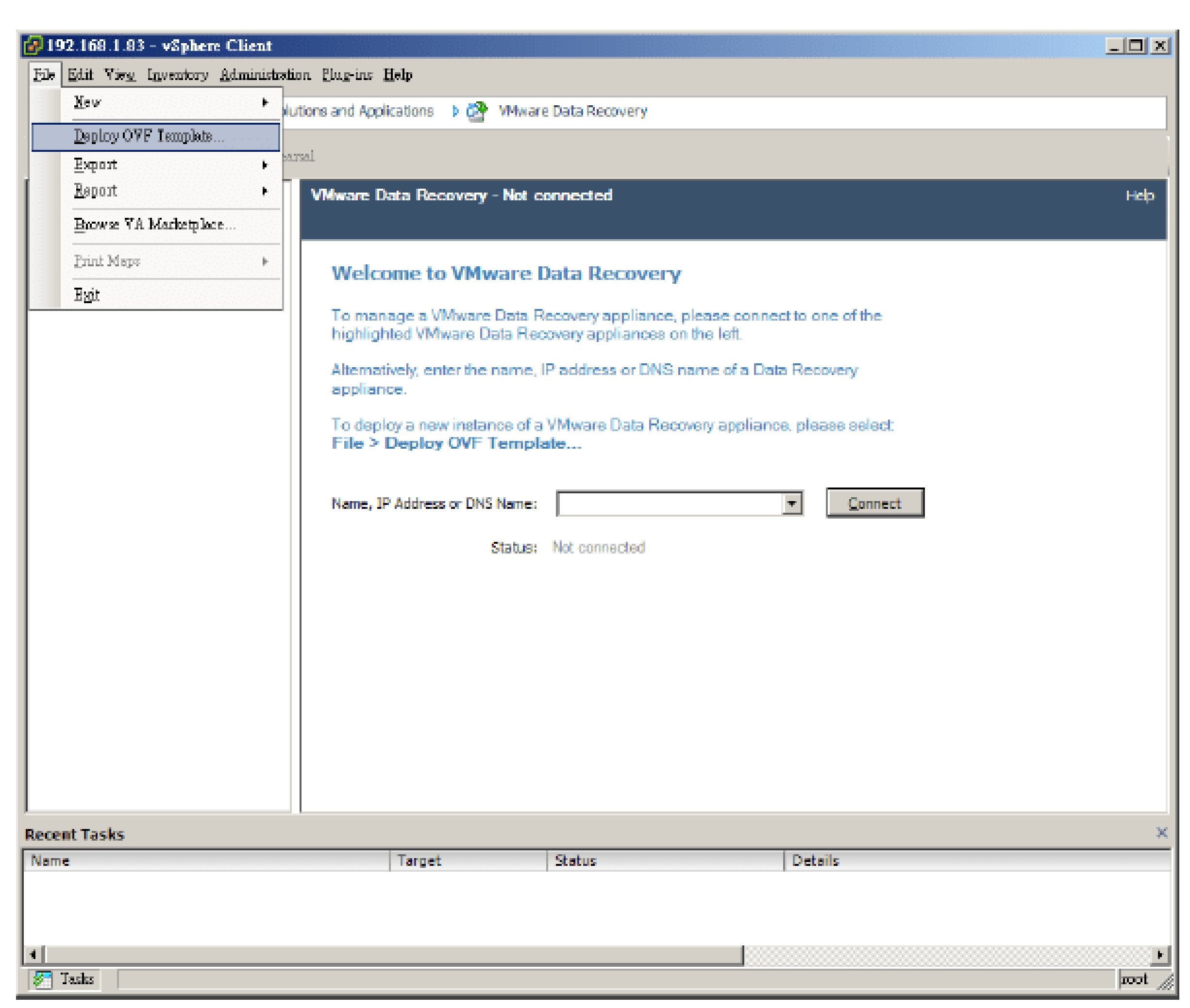
▲ 运行 VDR

2. 弹出如下图所示的画面，此时要做的就是将 VDR 的 VA 导入。



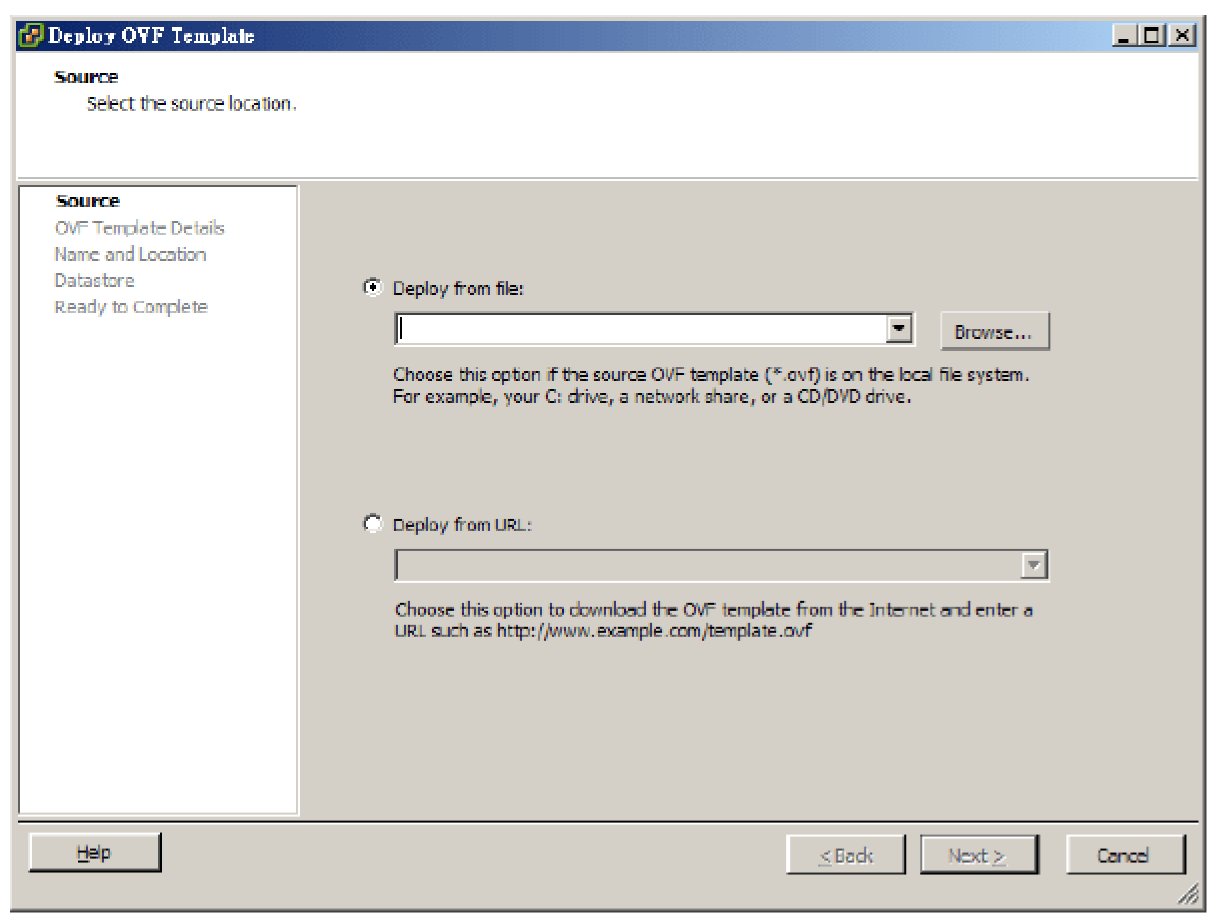
▲ 要求我们导入 VDR 的 VA

3. 选择 File/Deploy OVF Template 选项。



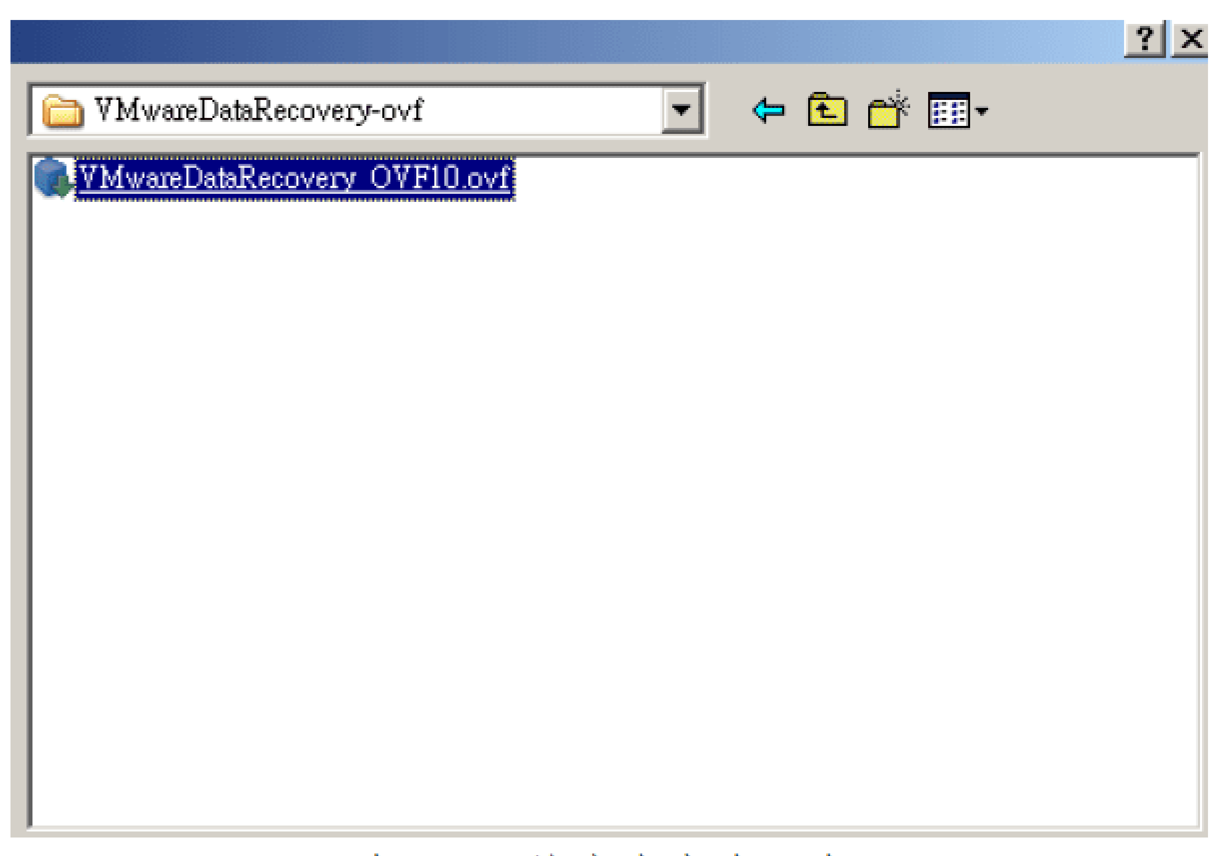
▲ 选择导入

4. 接下来弹出导入 OVF 的画面，单击 Browse 按钮。



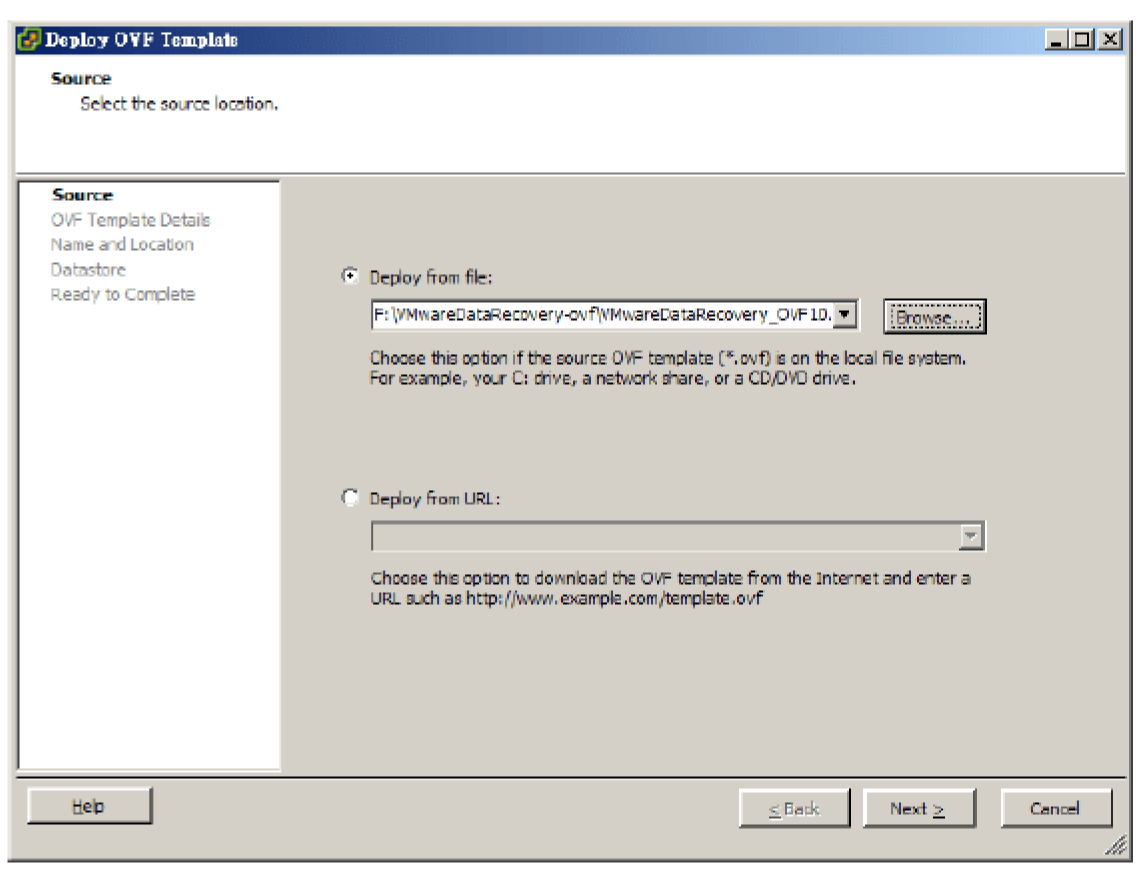
▲ VDR 是以 OVF 格式交付的

5. 进入 VDR 的安装光盘中，选择 VMwareDataRecovery-ovf 目录。选择其中的 OVF 文件。



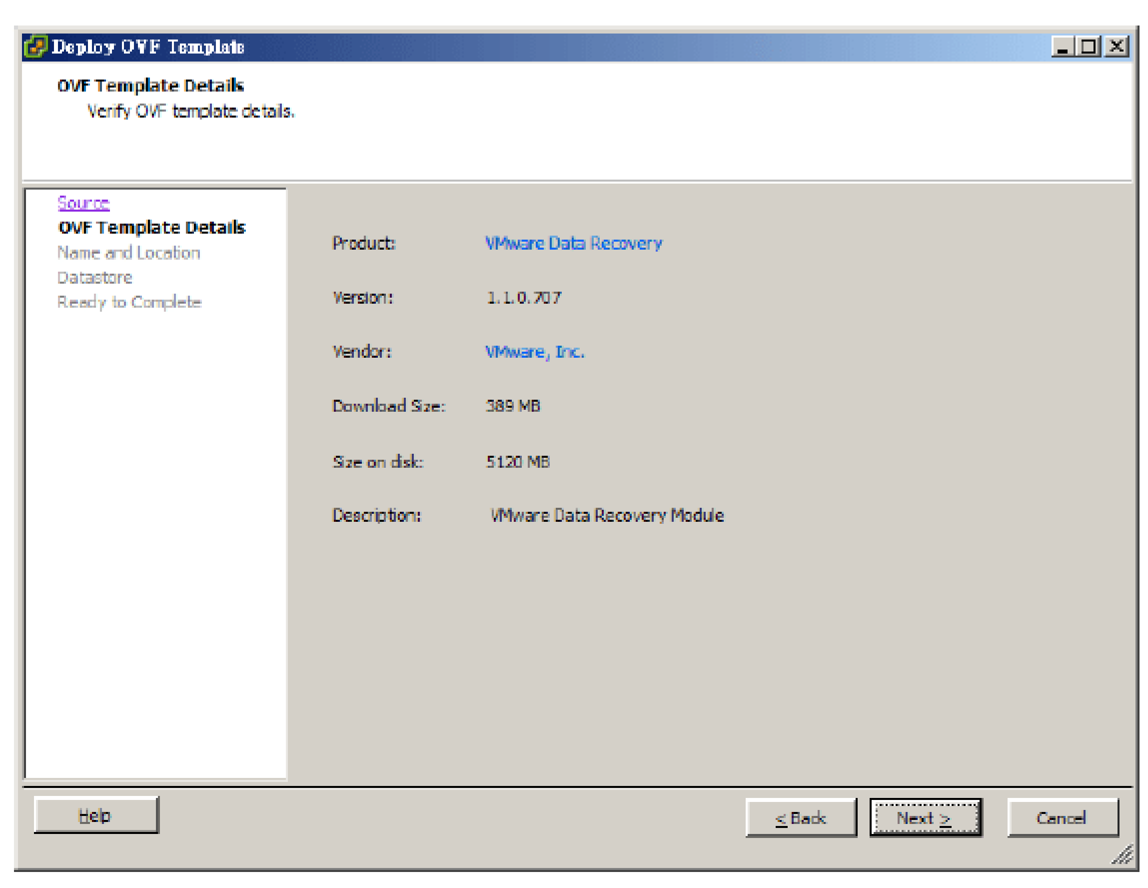
▲ 在 VDR 的光盘中有一个 OVF

6. 接下来单击 Next 按钮继续。



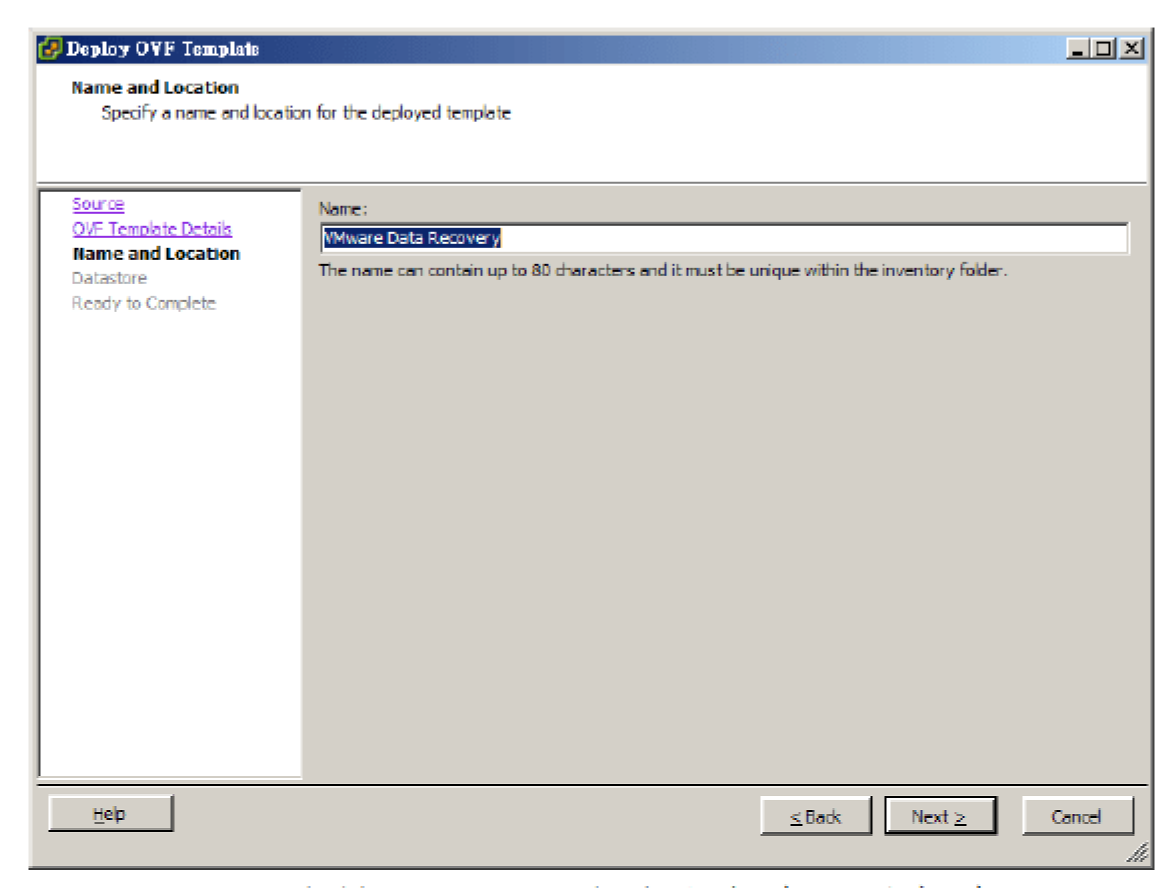
▲ 开始导入

7. 接下来会进入 VDR 的总结画面，单击 Next 按钮继续。



▲ 总结画面

8. 替这个 VM 命名，就以默认的 VMware Data Recovery 为名不要变动。单击 Next 按钮继续。

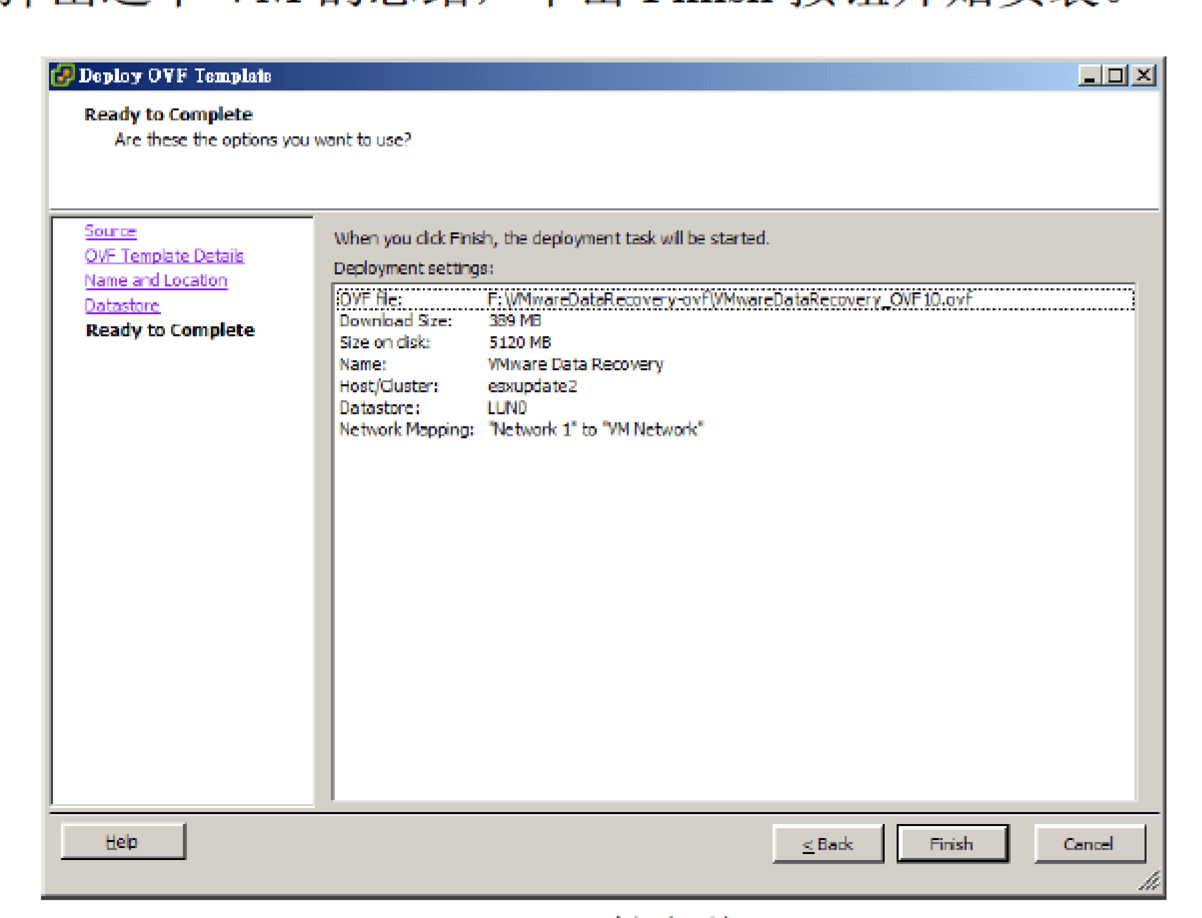


▲ 这是 VMware 的默认名称不要变动

9. 选择这个 VM/VA 的地点，就以 LUN 为主。单击 Next 按钮继续。

▲ 放置这个 VA 的地方

10. 接下来会弹出这个 VM 的总结，单击 Finish 按钮开始安装。



▲ 开始安装

11. 此时会开始将这个 VA 导入。

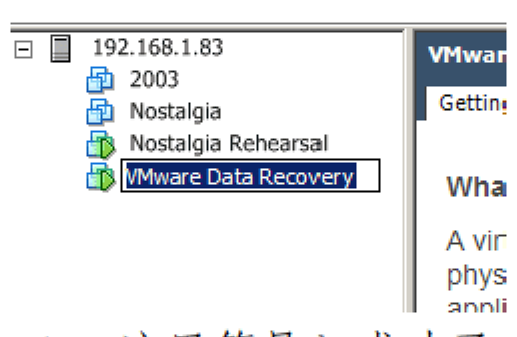
12. 当弹出如下图所示的画面时，意味着导入落实。



▲ 开始导入

▲ 导入落实

13. 此时回到 vSphere Client，会看到已经有一个 VM，就是刚才导入的 VDR，算导入成功了。但我们还要配置才能使用。



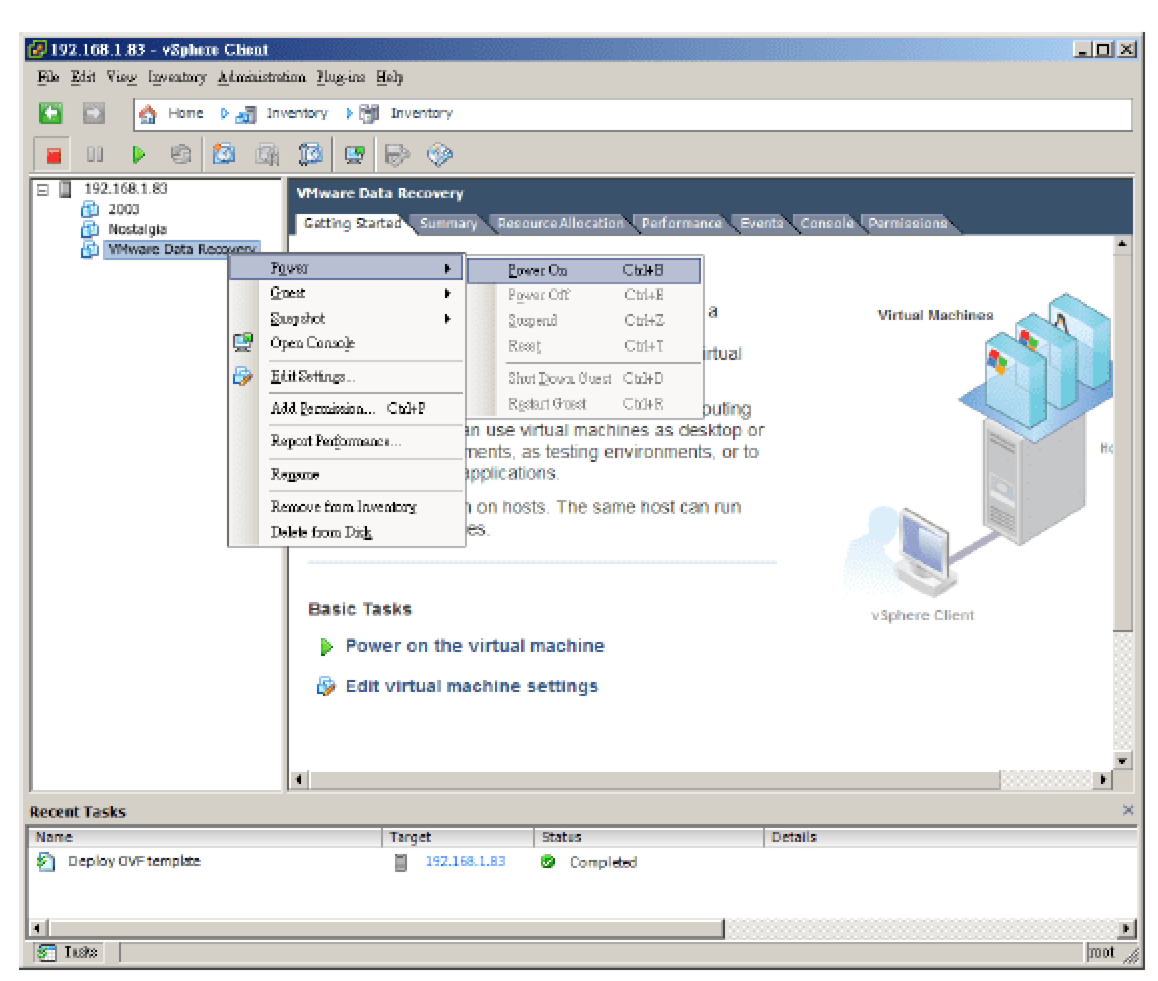
▲ 这里算导入成功了

3. 配置 VDR 这个 VA 的网络参数

当我们运行 vSphere Client 中的 VDR VA 时，事实上系统会要求我们连接到这个 VA，不管是使用 IP、名称或是 DNS 名称，因此我们至少要将这个 VA 引导，并且配置 IP。前面说过，一般的 VA 都是通过网页界面来配置，因此我们必须知道这个 VA 的 IP 以及通信端口号。下面就是配置这个 VA 的步骤。

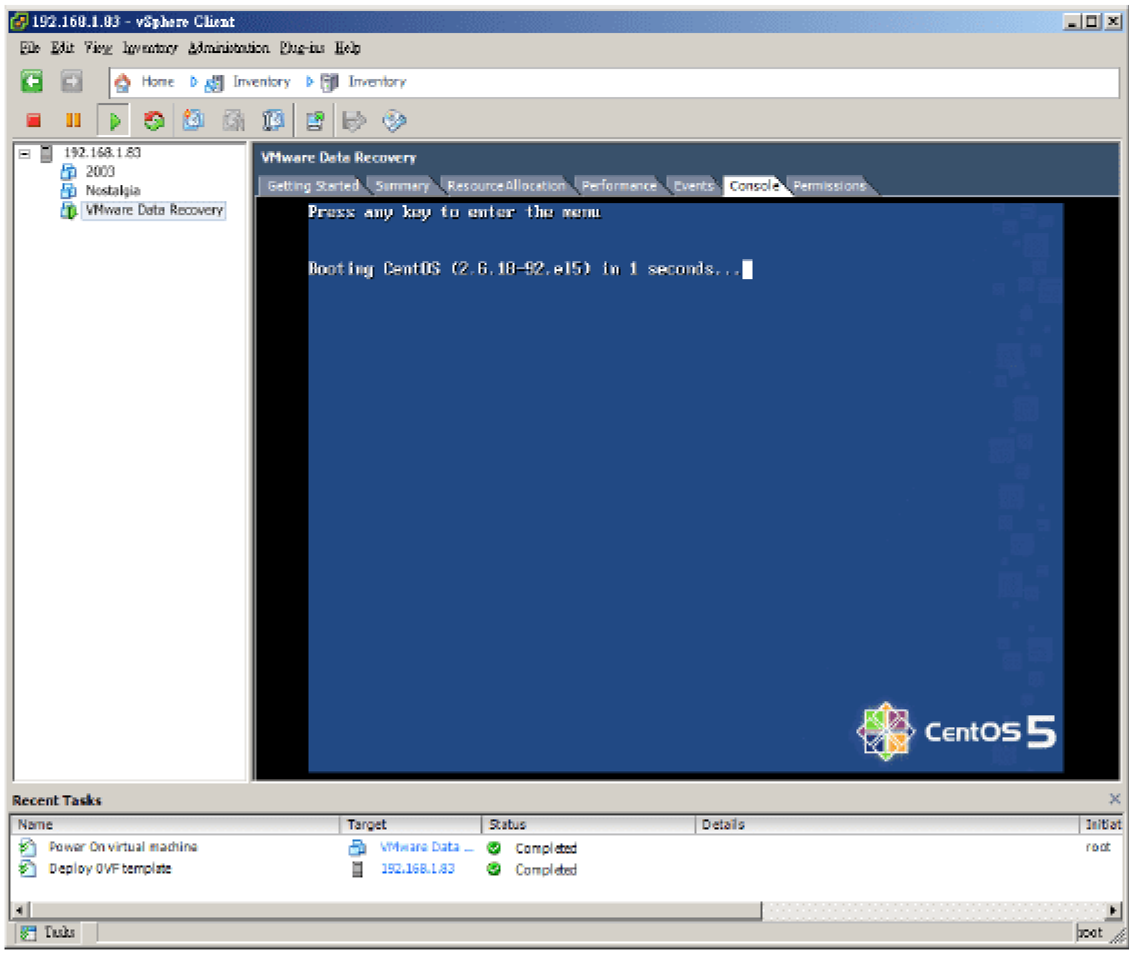
► 配置 VDR VA 的网络参数

1. 进入 vSphere Client 的画面，并且选择将 VMware Data Recovery 这个 VA 引导。



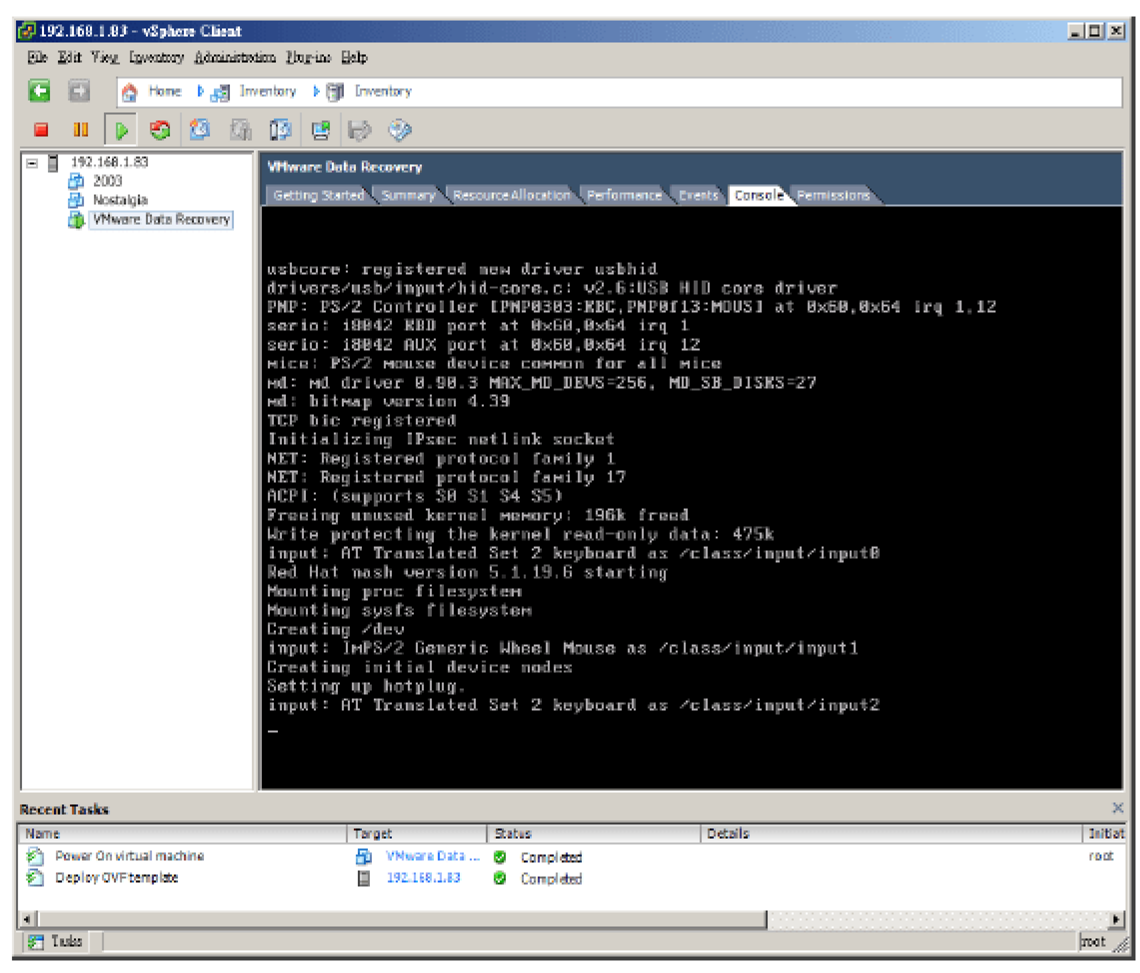
▲ 将这个 VA 引导

2. 接下来会进入 VDR 这个 VA 的画面，你会发现这是一个 CentOS Linux 的系统。



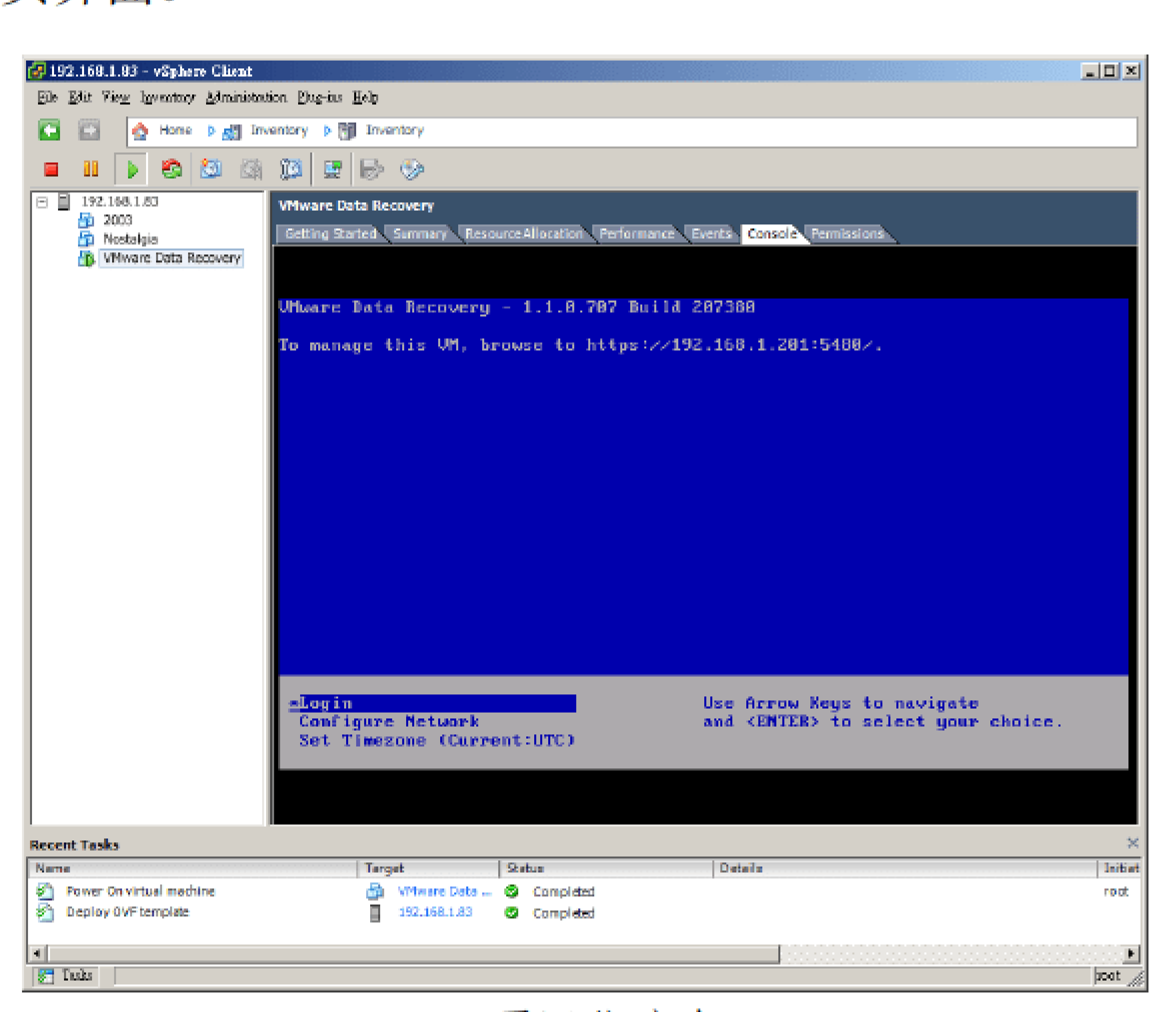
▲ 这个 VA 是一个 Linux 系统

3. 接下来会进入标准 Linux 的引导画面。



▲ 标准的 Linux 引导画面

4. 引导后我们可以看到这个 VM 的 IP 是 https://192.168.1.201:5480，这就是我们要配置这个 VA 的网页界面。



▲ 要记住这个 IP

5. 打开浏览器，并且进入上面的 IP 地址。





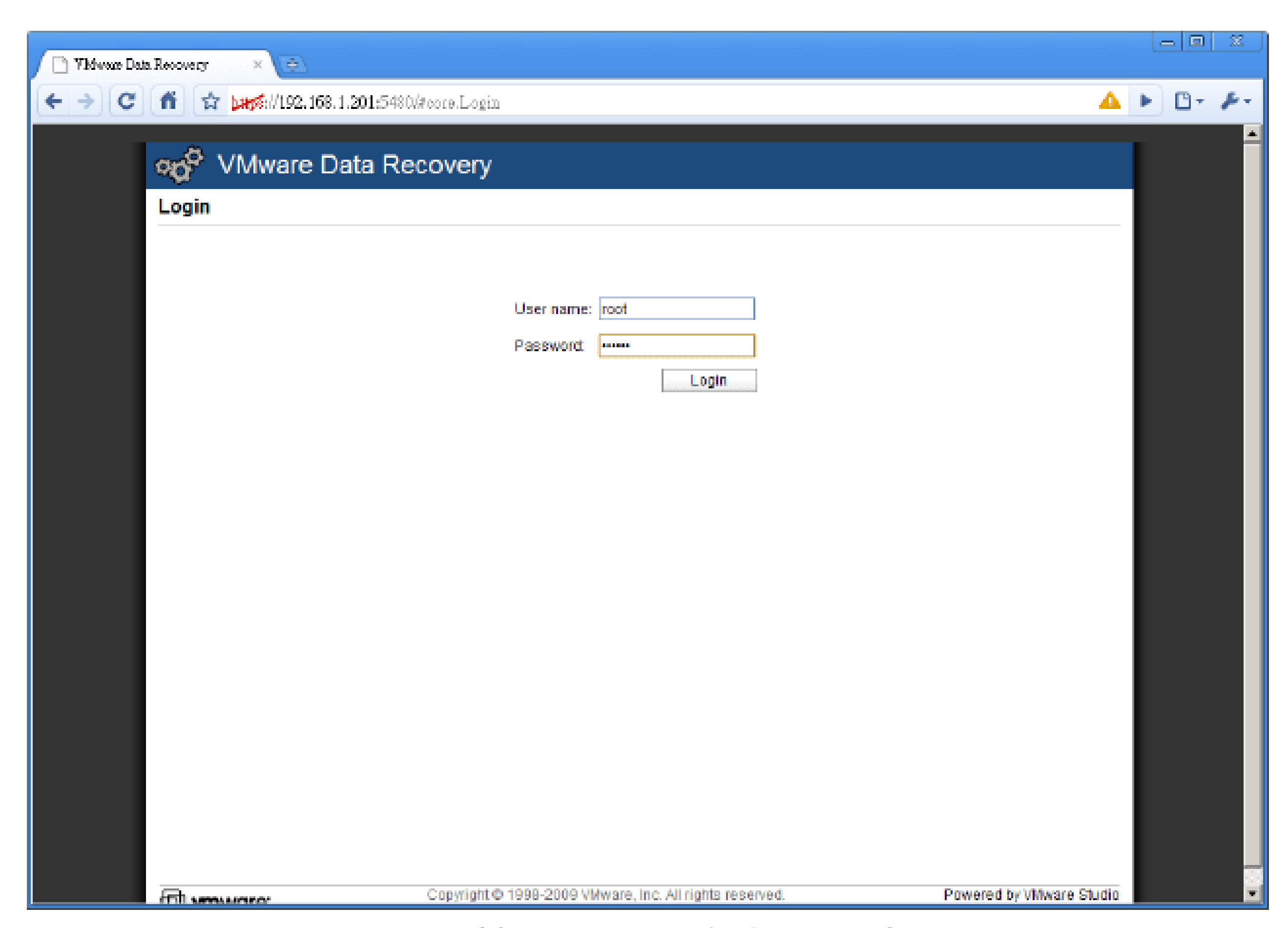
▲ 进入这个地址

6. 此时会弹出 https 的安全报警。单击“仍然继续”按钮。



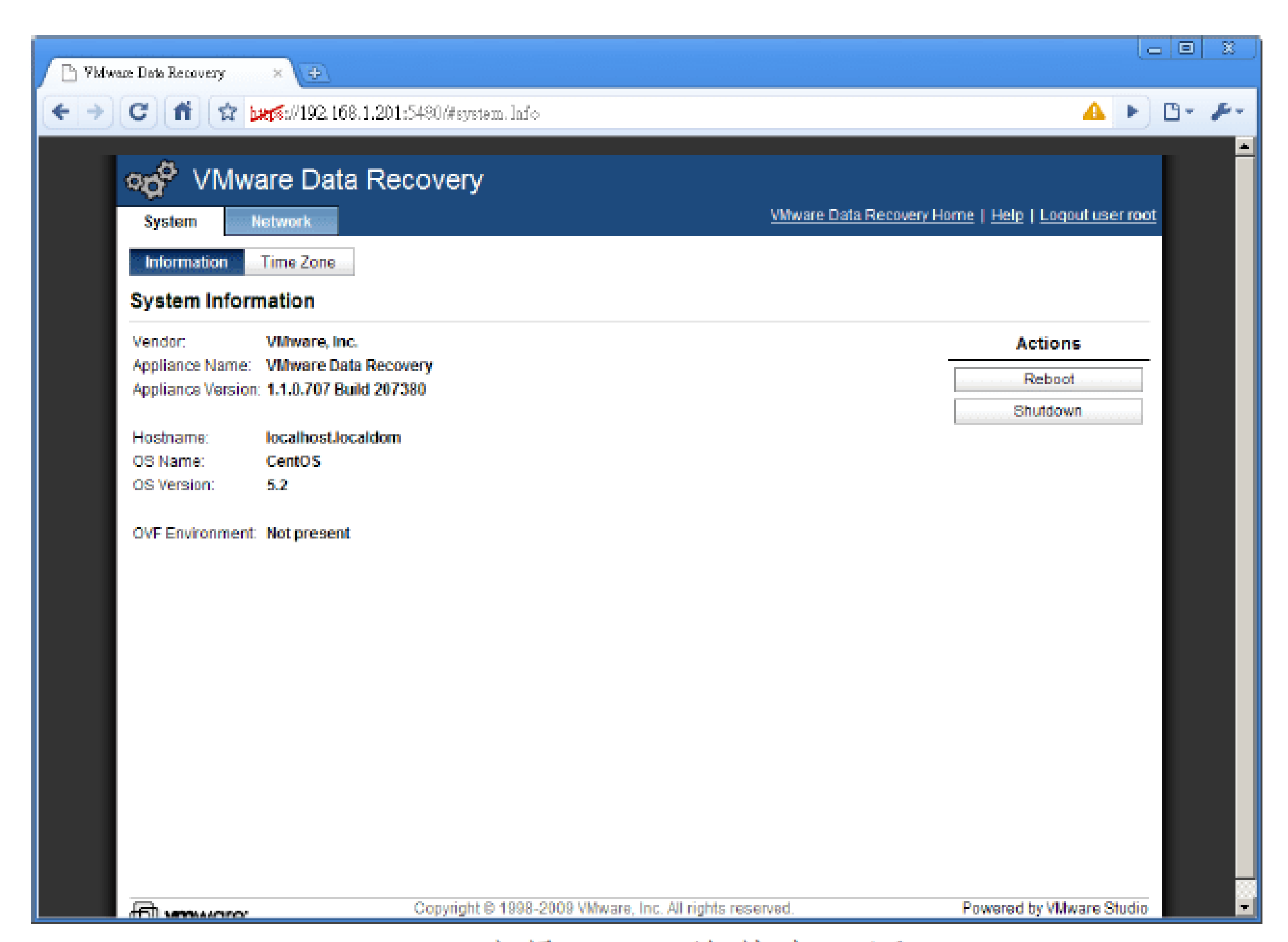
▲ 继续进入

7. 接下来键入默认的账号口令 root 以及 vmw@re。



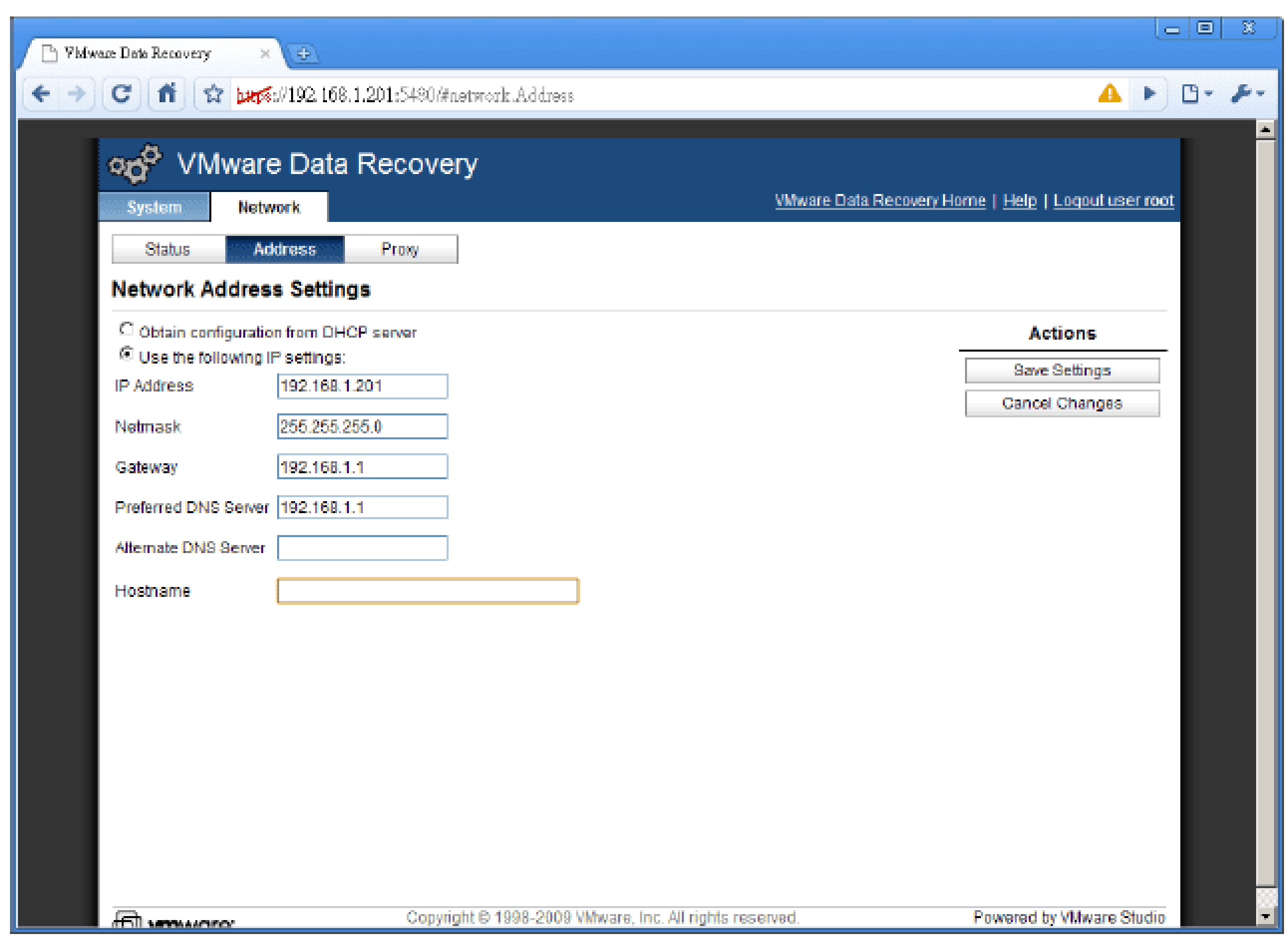
▲ 键入 VDR 的账号口令

8. 登录之后，就是简单的配置，选择 Network/Address 选项。



▲ 这是 VDR 的基本配置

9. 键入 IP 之后就不需要使用 DHCP，因为有时使用 DHCP 来配置会有风险。配置好 IP 之后别忘了单击 Save Settings 按钮存盘。



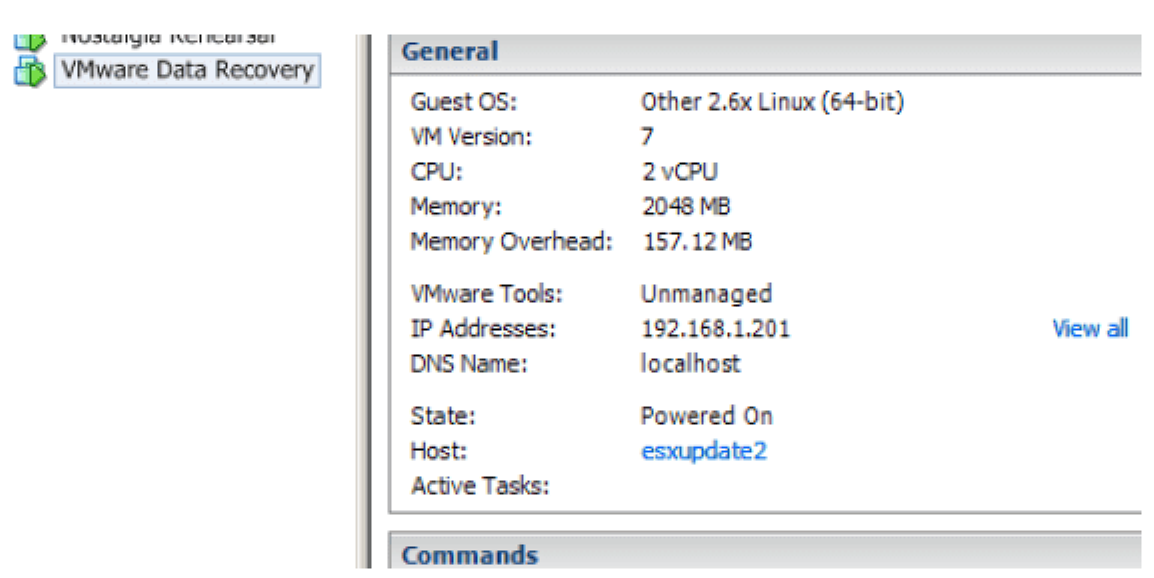
▲ 不要使用 DHCP

4. 激活 VDR 这个 VA 功能

当所有的参数配置好之后，就必须在 vSphere Client 中激活 VDR 功能了。激活 VDR 就是让运行 VDR 的 VA 引导，并且在 vSphere Client 中连接并且运行功能，下面我们就来看看。

▶ 激活 VDR 功能

1. 首先我们进入 vSphere Client，将 VDR 引导。
2. 在弹出的画面时，意味着 VDR 已经引导成功。



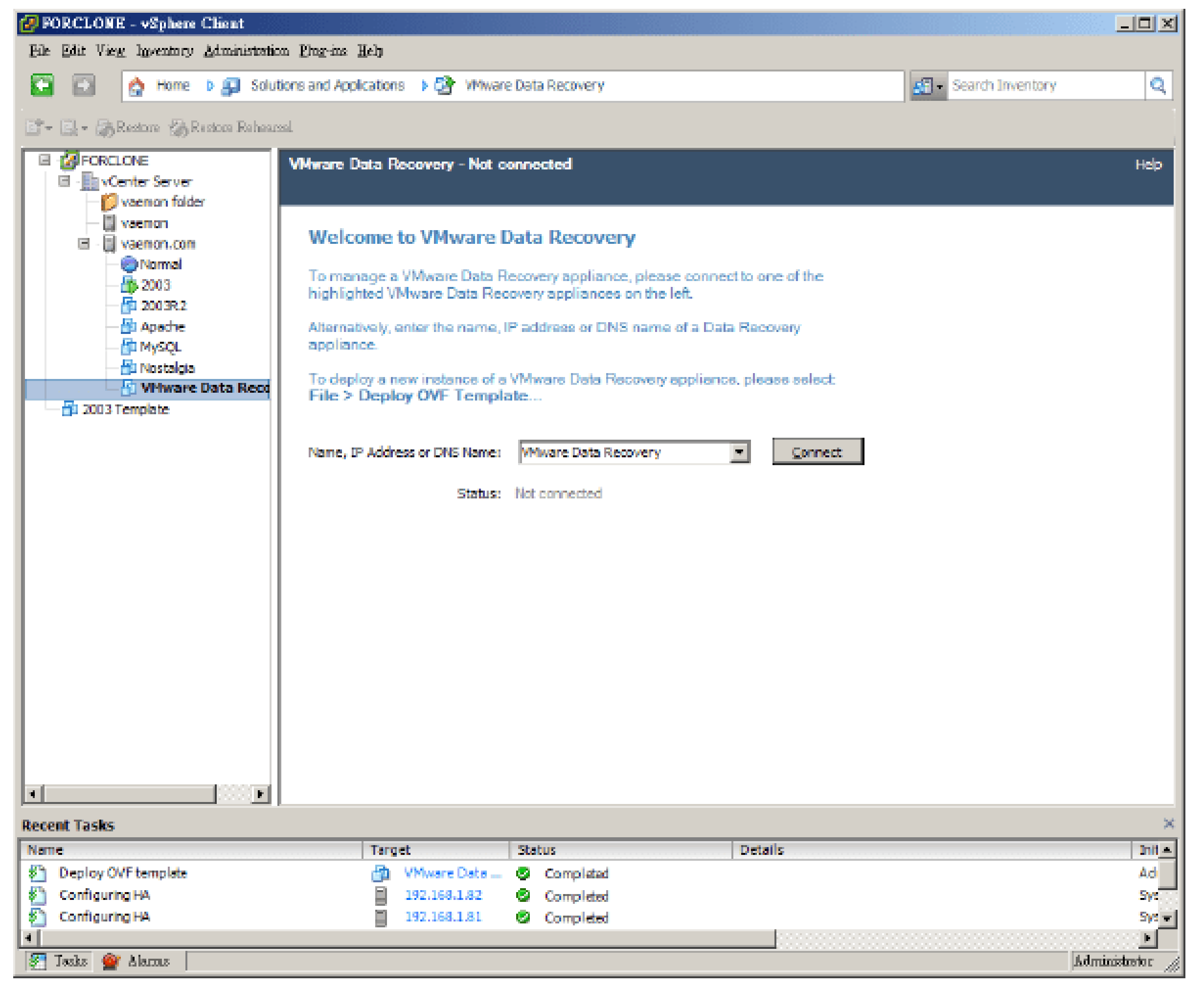
▲ 引导成功

3. 此时进入 Home 并且选择 VMware Data Recovery 的功能。



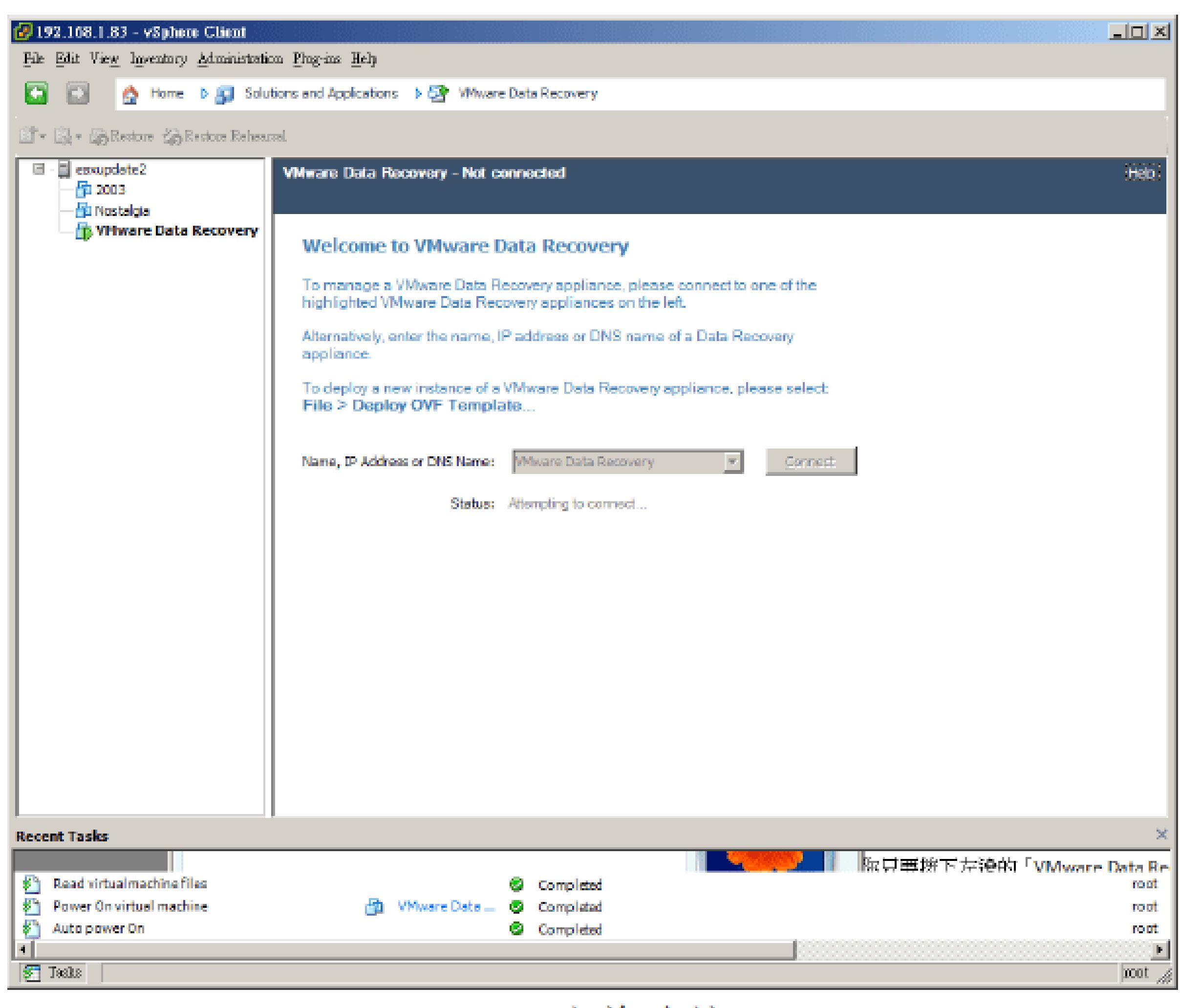
▲ 开始运行 VDR

4. 此时系统会要求我们联机，你只要选择左边的 VMware Data Recovery 目录，这个 VA 就会自动弹出在中间的图示，单击 Connect 按钮即可。



▲ 直接联机到这个 VA 中

5. 此时系统会试着连接。



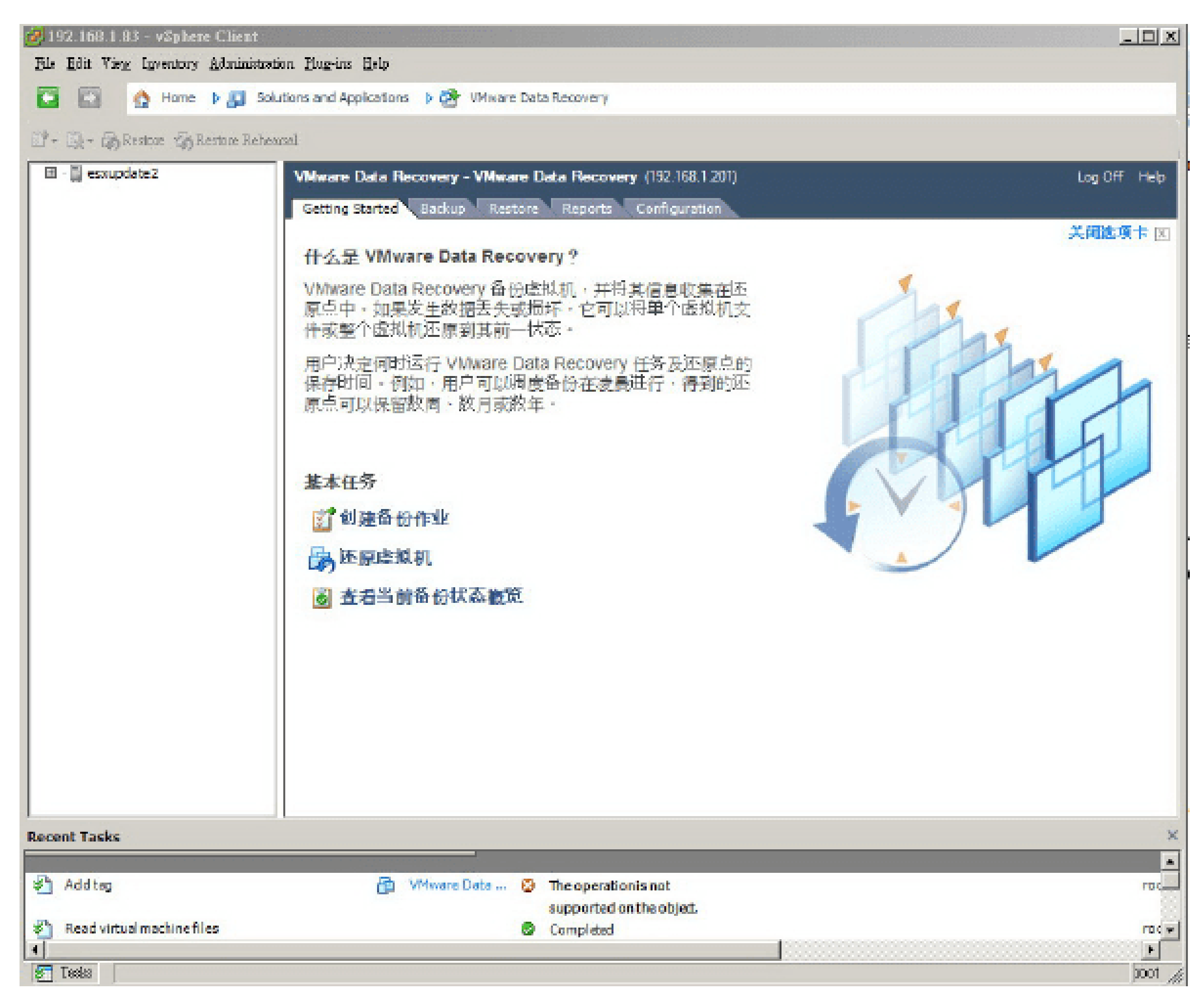
▲ 试着连接

6. 接下来会弹出键入运行 VDR 那台 ESX 主机的账号口令，键入账号口令。



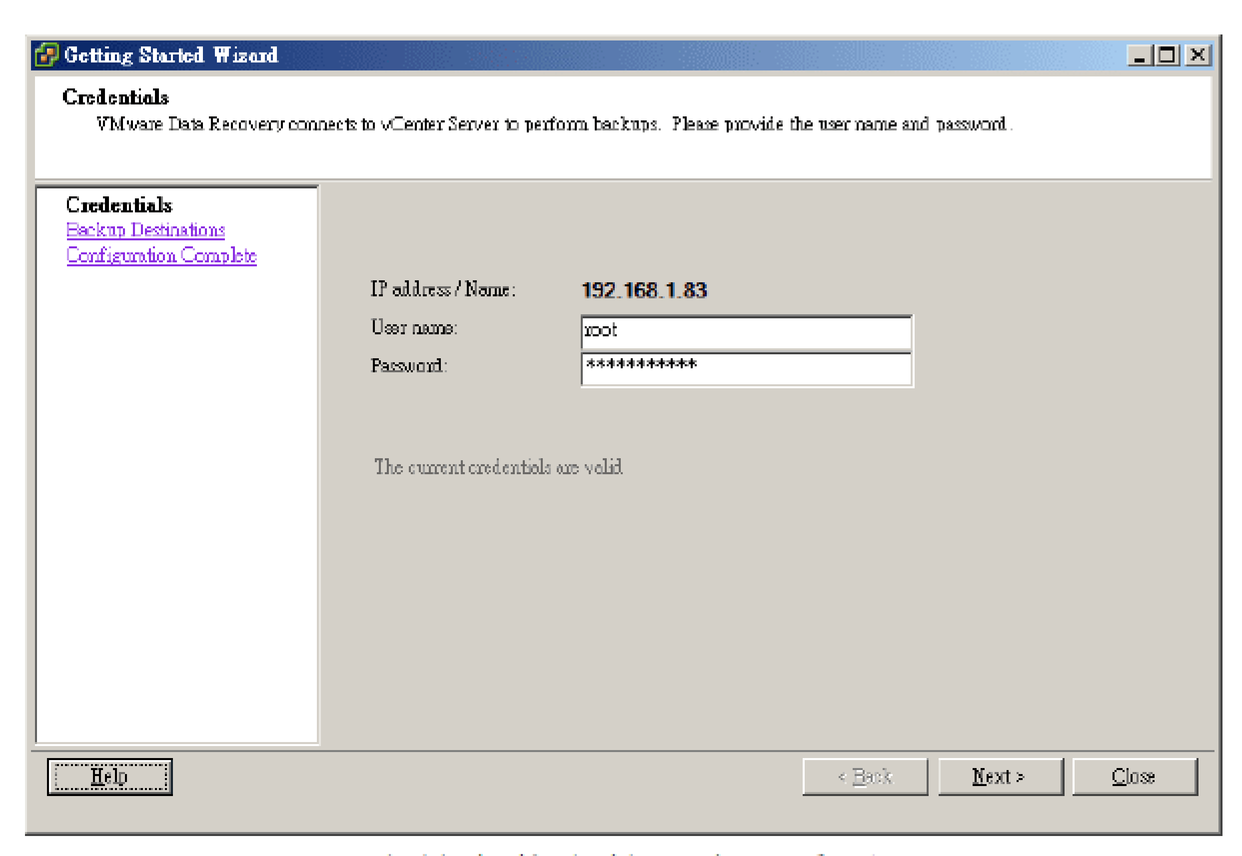
▲ 键入 ESX 的账号口令

7. 接下来你会发现在 vSphere Client 中的 VDR 已经激活。



▲ VDR 激活了

8. 此时会直接进入 VDR 的配置画面。要求你键入安装这台 VDR ESX 主机的账号口令。由于我们还没有添加备份目的，因此先结束这个向导。



▲ 这是备份向导，先不需要运行

21.3.2 配置 VDR

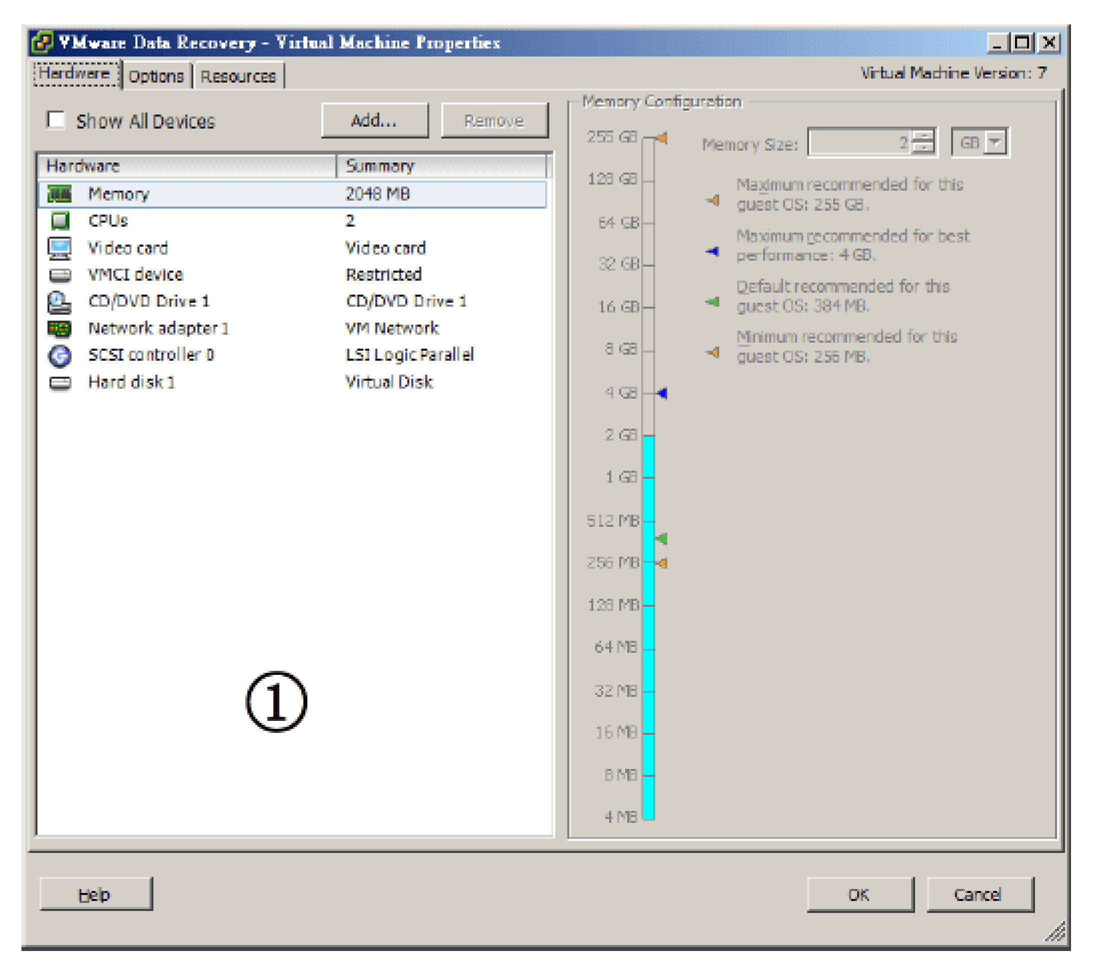
当安装好 VDR 之后，我们还无法用来备份 VM，主要的原因是我们还需要配置备份的目的及磁盘等。我们就来看看如何配置。

1. 替 VDR 添加一个虚拟磁盘

备份 VM 的目的可以有很多种，一般的我们可以替 VDR 这个 VA 添加一个虚拟磁盘，然后将 VM 备份进来。添加备份磁盘的步骤很简单，就是替这个 VM 新建一个磁盘，下面就是完整步骤。

▶ 替 VDR 新建一个虚拟磁盘

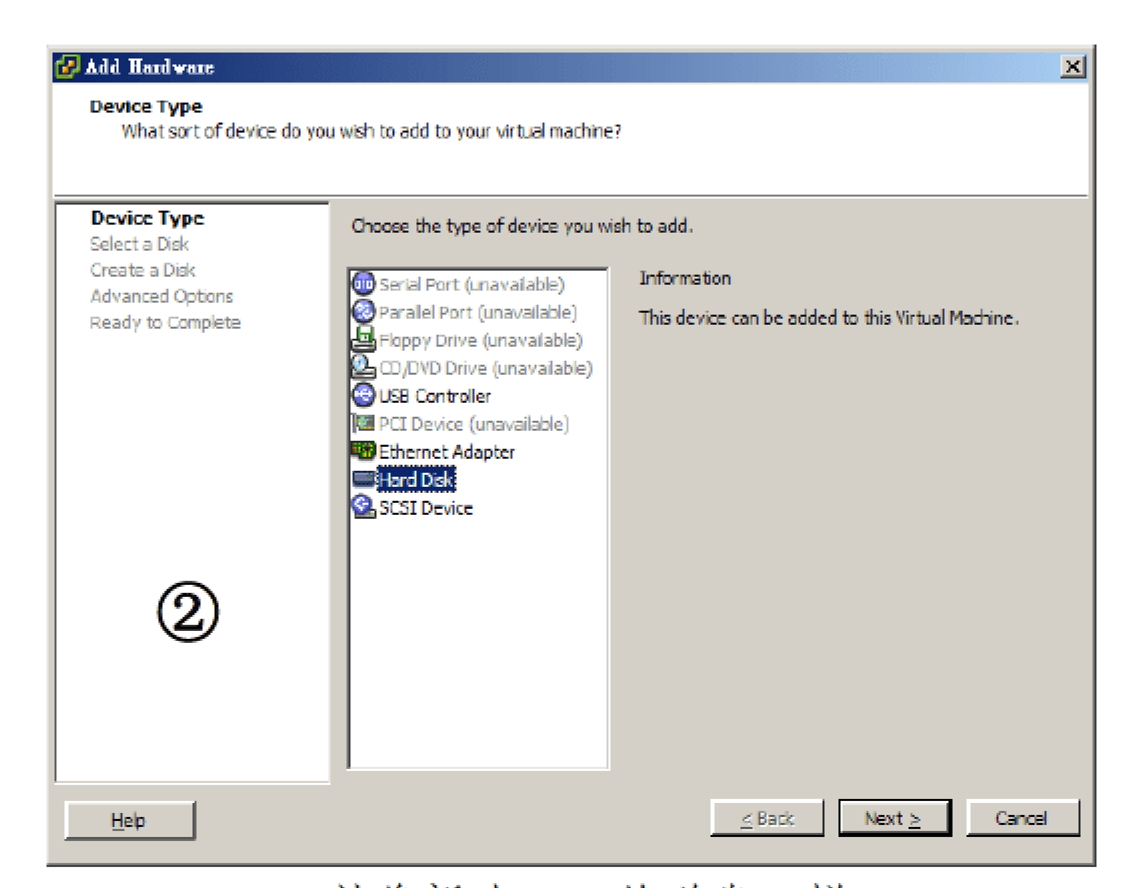
1. 进入这个 VA 的配置项。并且单击 Add 按钮。此时 VDR 是引导状态也没关系，因为 CentOS 是支持 Hot Swap 的。



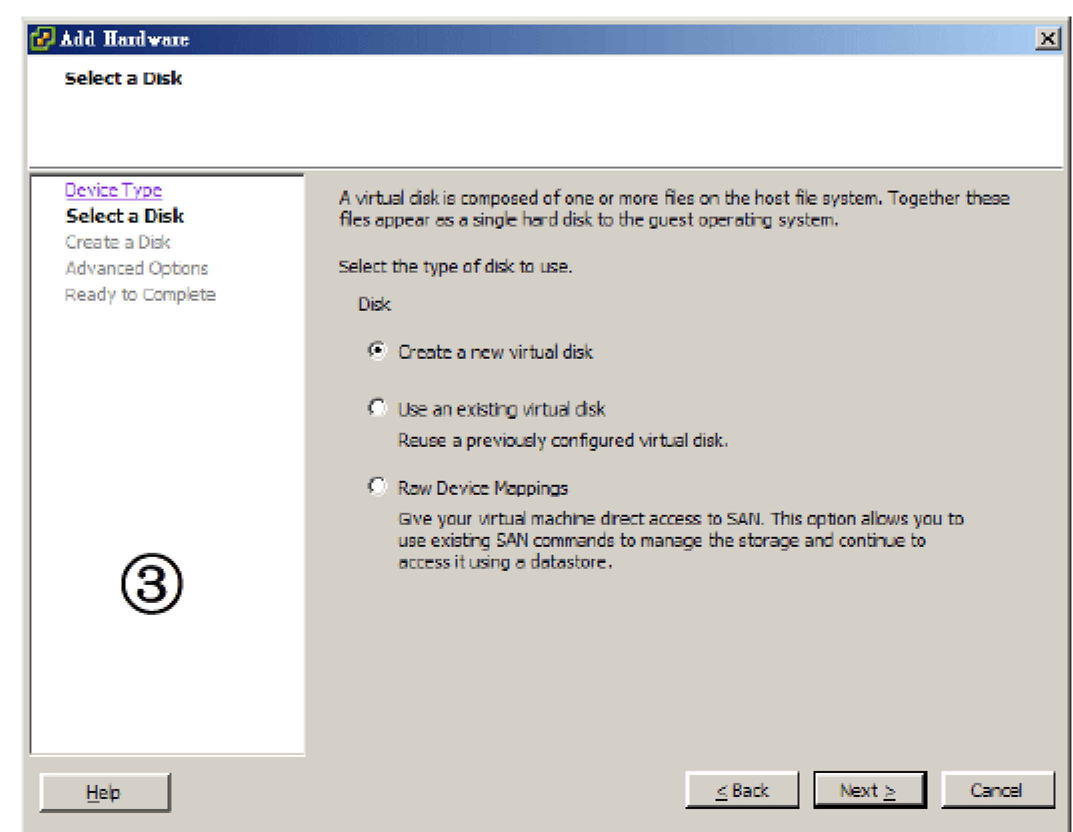
▲ 先替这个 VA 添加一个备份用的磁盘

2. 选择 Hard Disk，再单击 Next 按钮。

3. 选择创建新的虚拟磁盘，单击 Next 按钮。



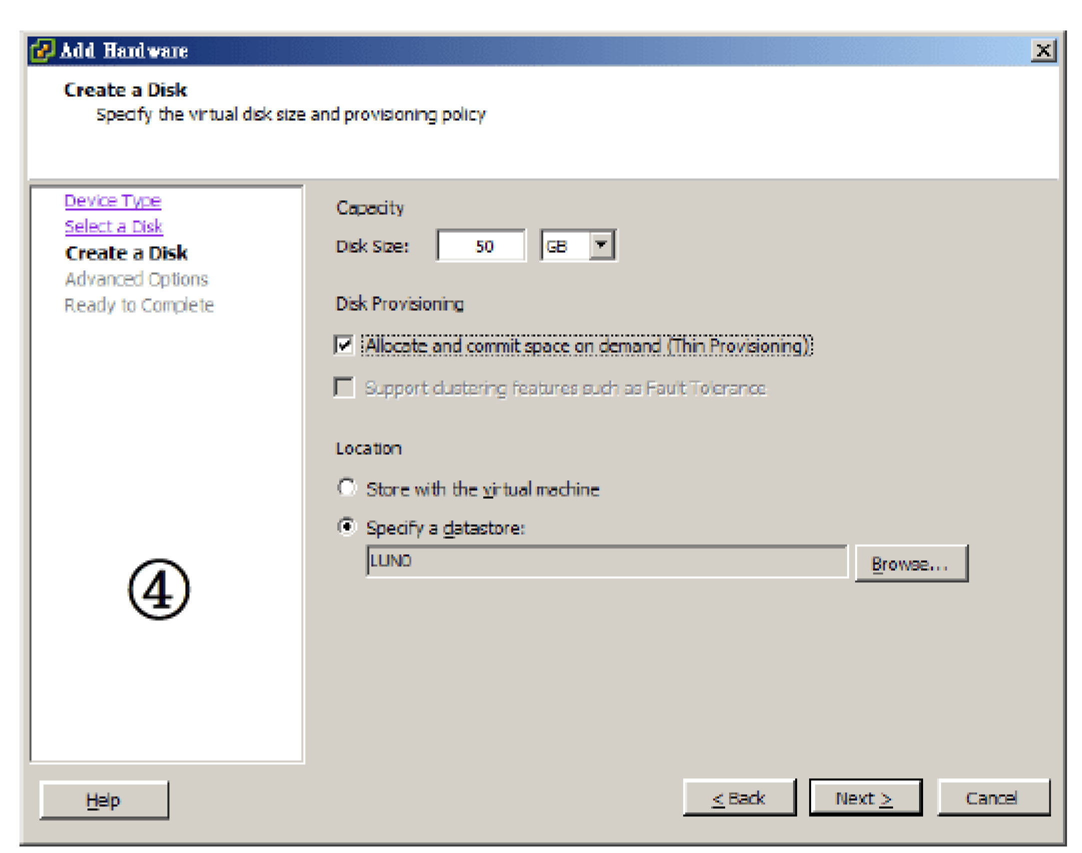
▲ 就像新建 VM 的磁盘一样



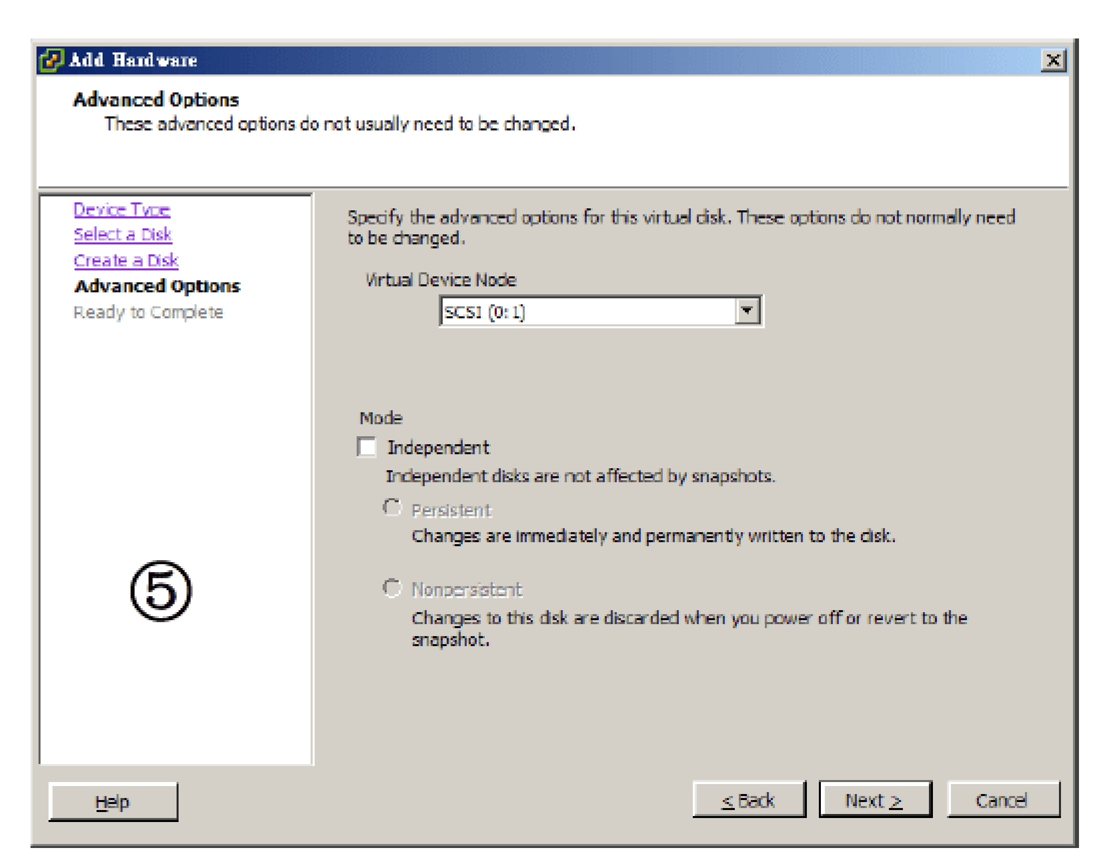
▲ 创建新的磁盘

4. 我们就选择 50GB，并且激活 Thin Provisioning，将这个虚拟磁盘放在 iSCSI 的 LUN0 上。

5. 接下来就是标准的 SCSI 配置。



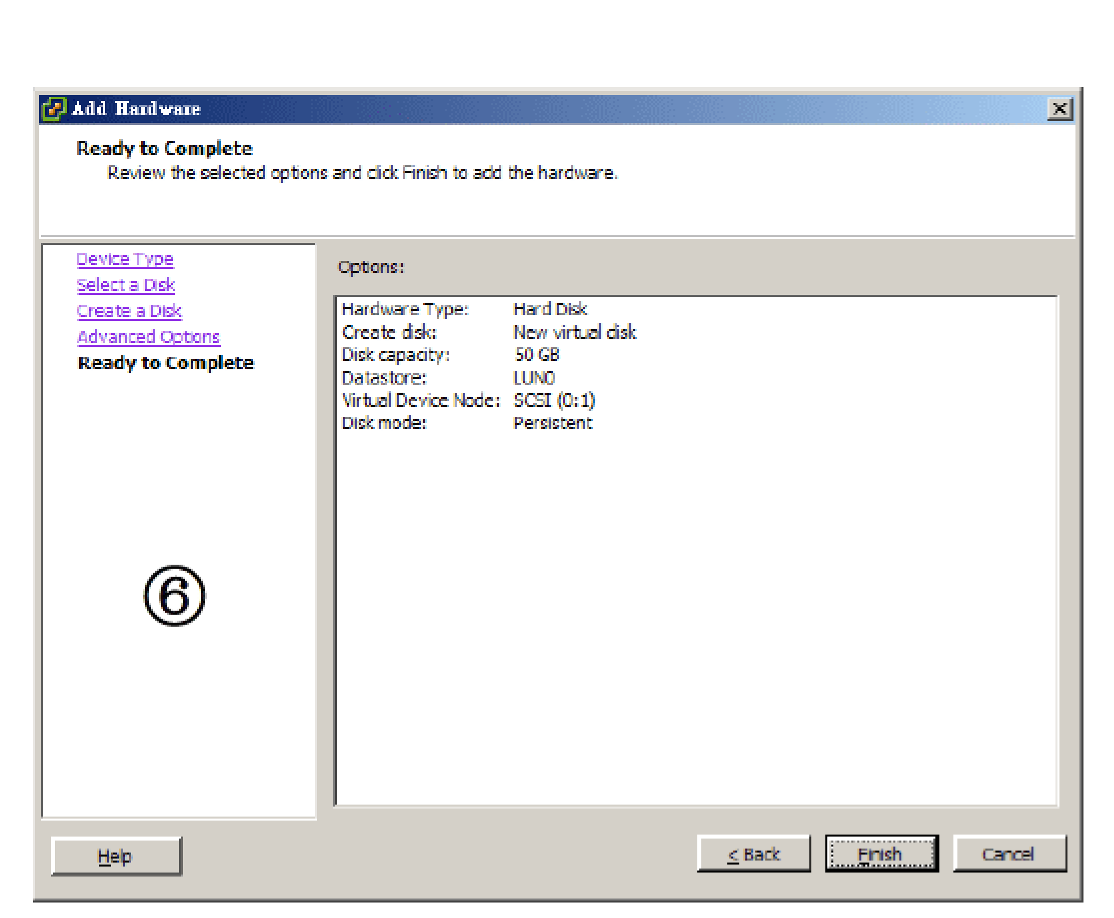
▲ 选择虚拟磁盘的参数



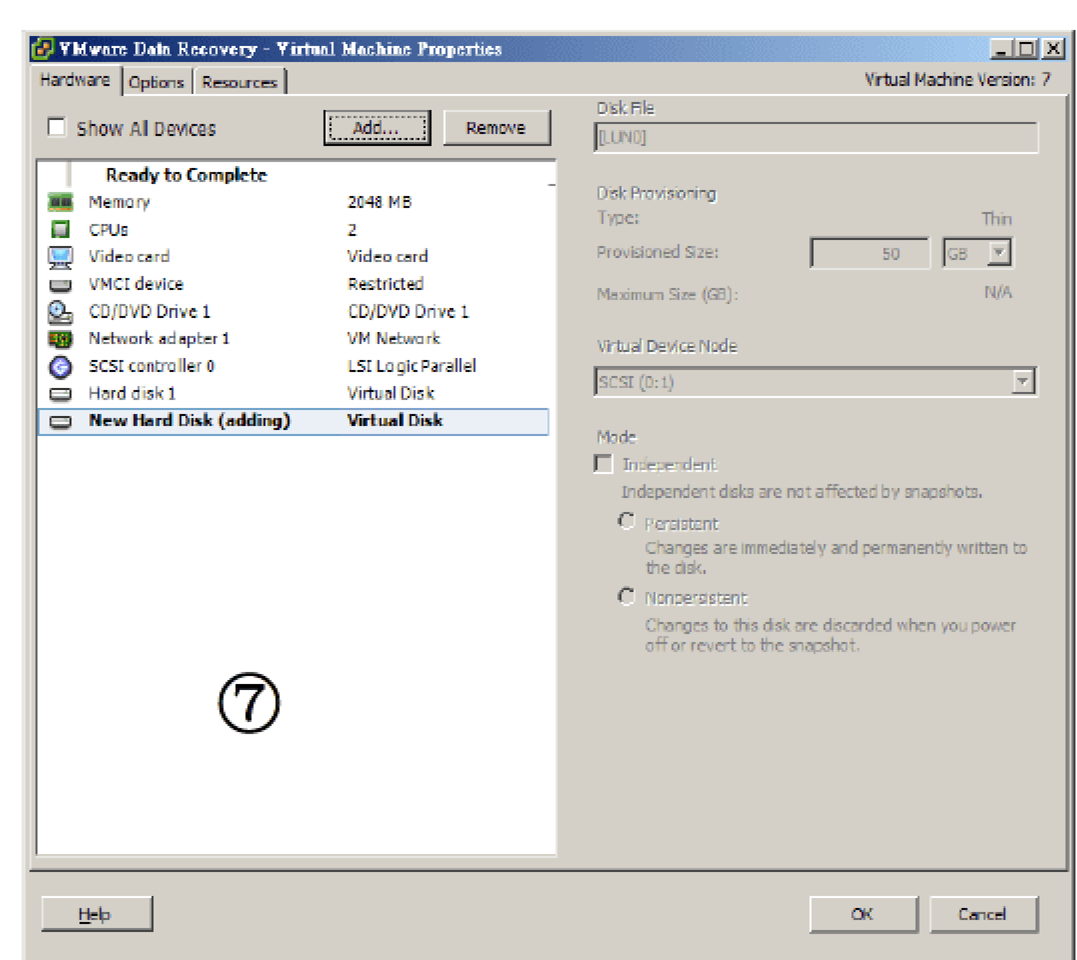
▲ 选择 SCSI 配置，会在之后的 VDR 画面弹出

6. 弹出总结画面后单击 Finish 按钮。

7. 最后可以看到这个备份用的虚拟磁盘已经创建好了。

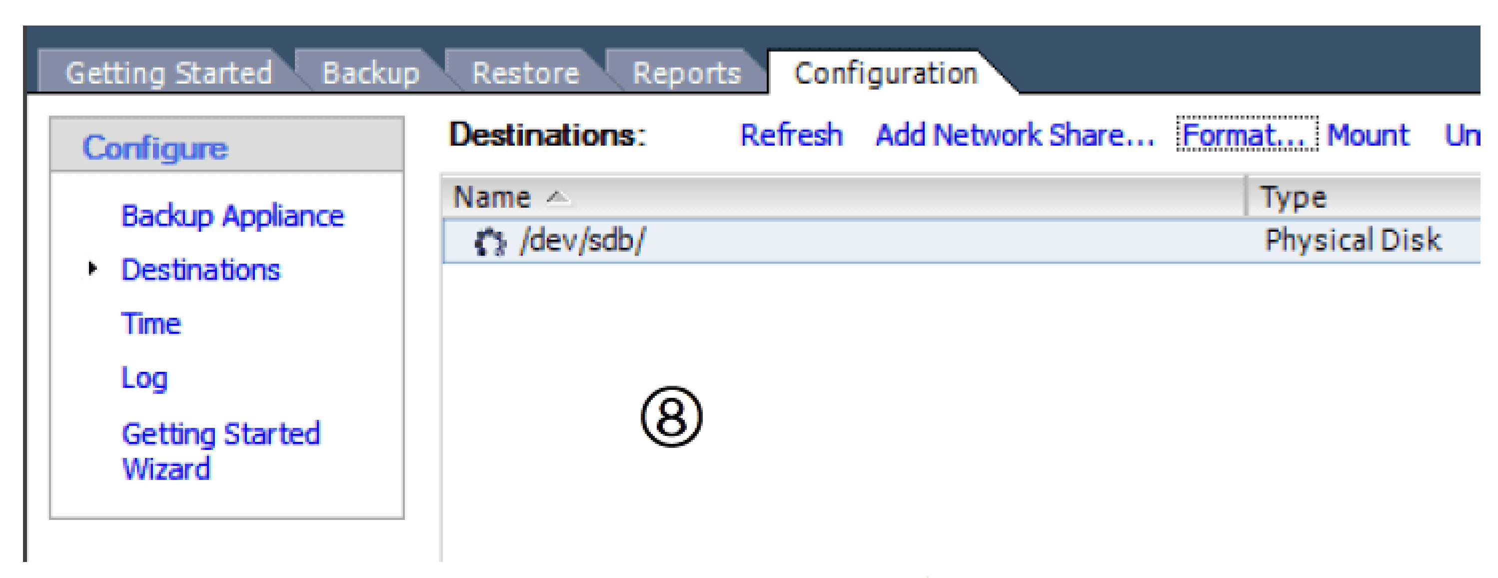


▲ 总结画面



▲ 加入成功

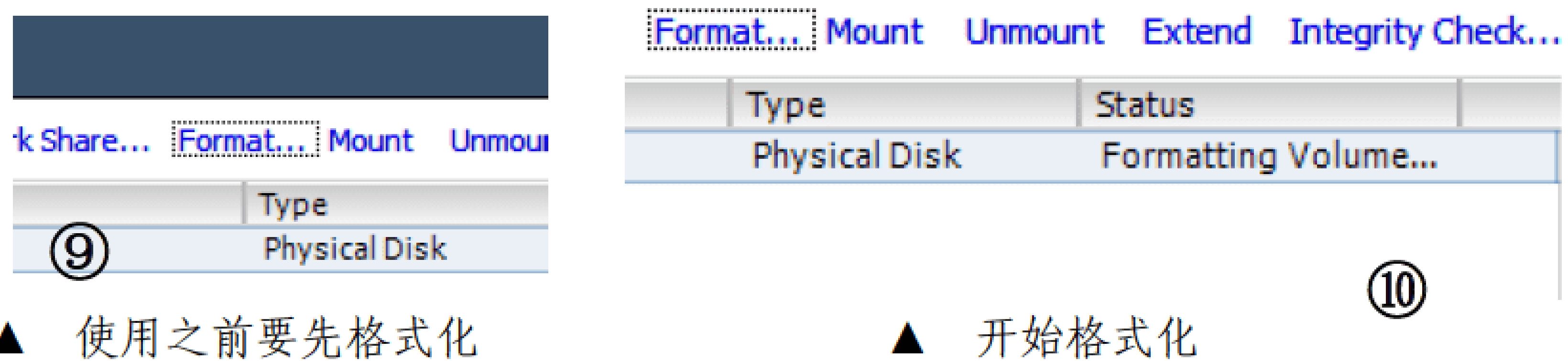
8. 此时我们进入 VDR，单击 Refresh，可以看到刚才加入的虚拟磁盘也弹出了。



▲ 刚才加入的磁盘弹出了

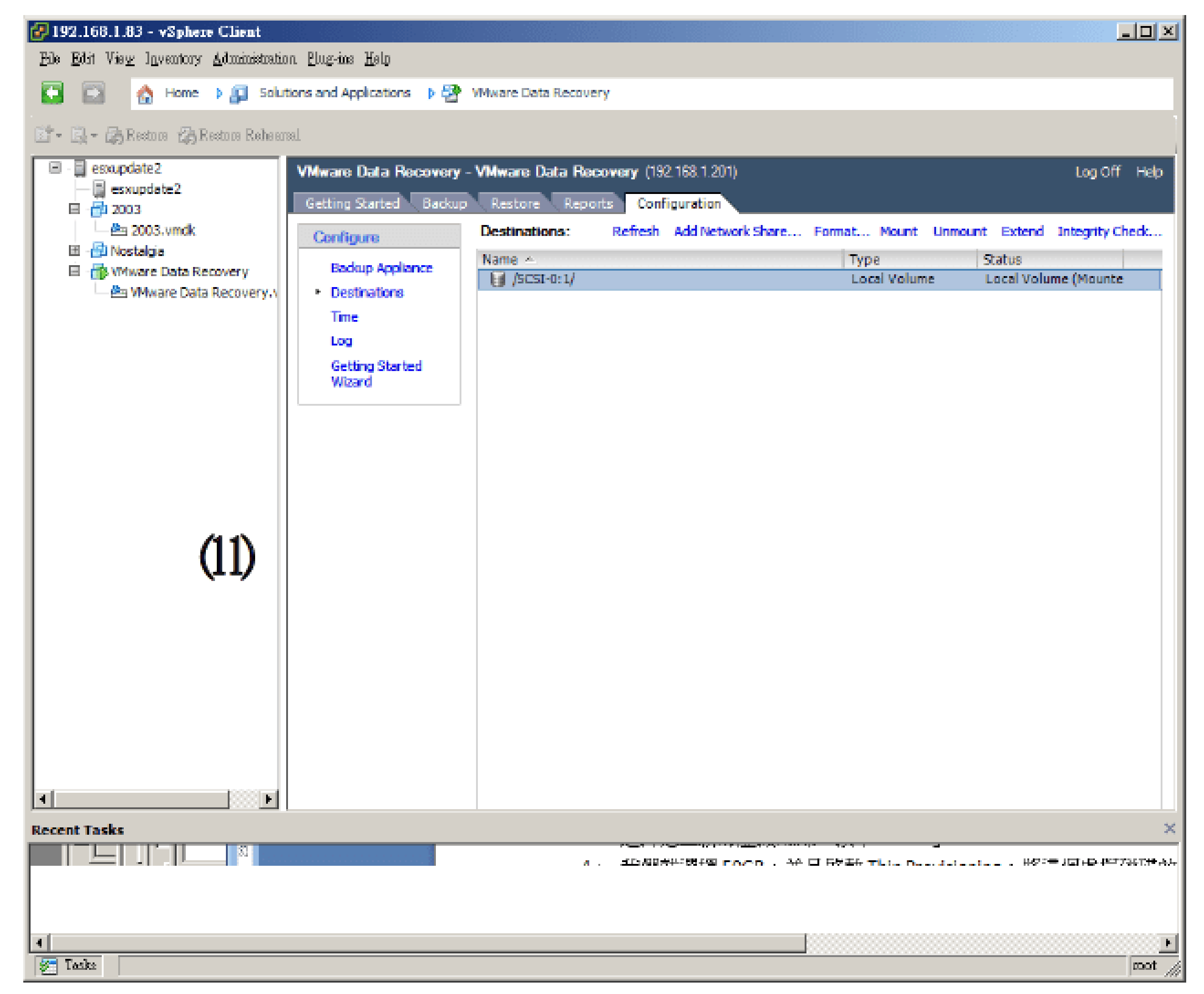
9. 接下来我们要先将这个硬盘格式化，单击 Format。

10. 系统会开始格式化这个硬盘。



▲ 使用之前要先格式化　　▲ 开始格式化

11. 当弹出如下图所示的画面时，意味着格式化完毕，这个虚拟硬盘也可以作为备份的目的了。



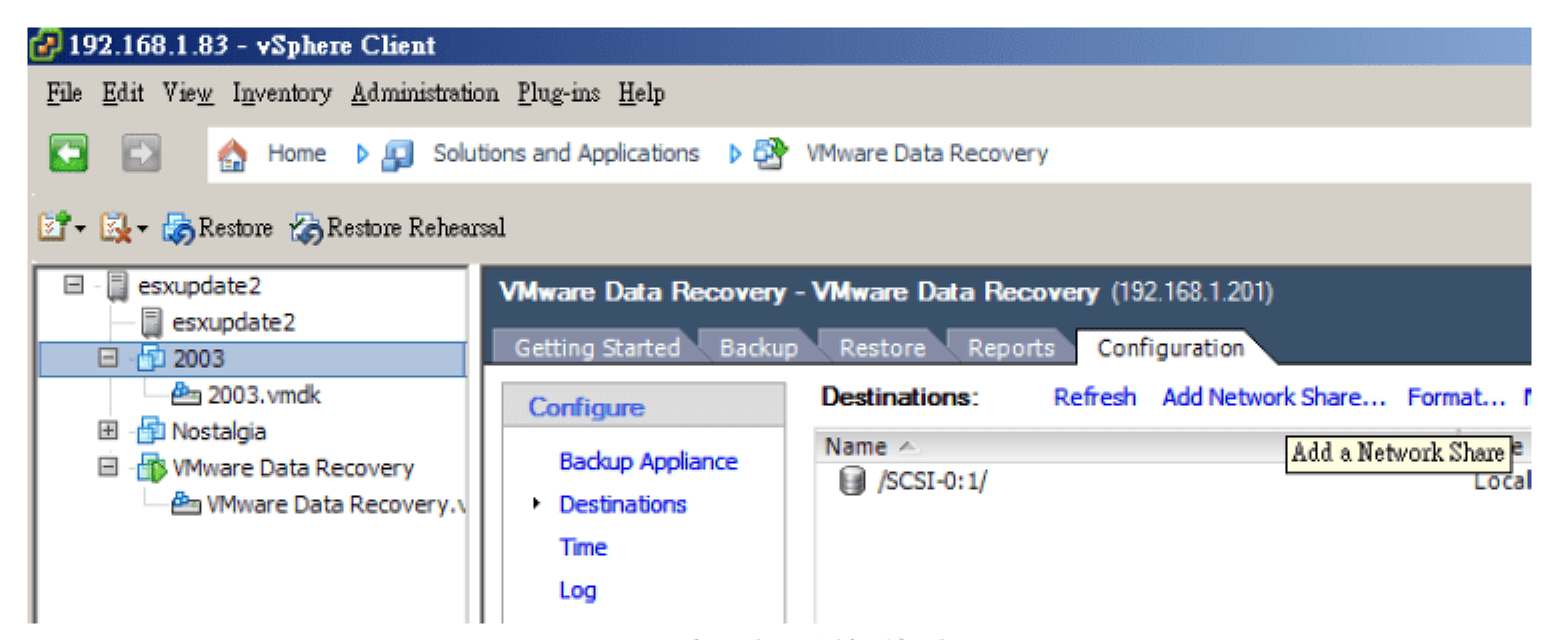
▲ 可以使用了，这个虚拟磁盘也挂上 VDR 中

2. 添加网络共享设备备份

大部分的情况下，我们会将独立存储拿来作为备份的目的。VDR 支持 NFS 的备份，我们就来看看如何替 VDR 加入一个 NFS 作为备份目的。

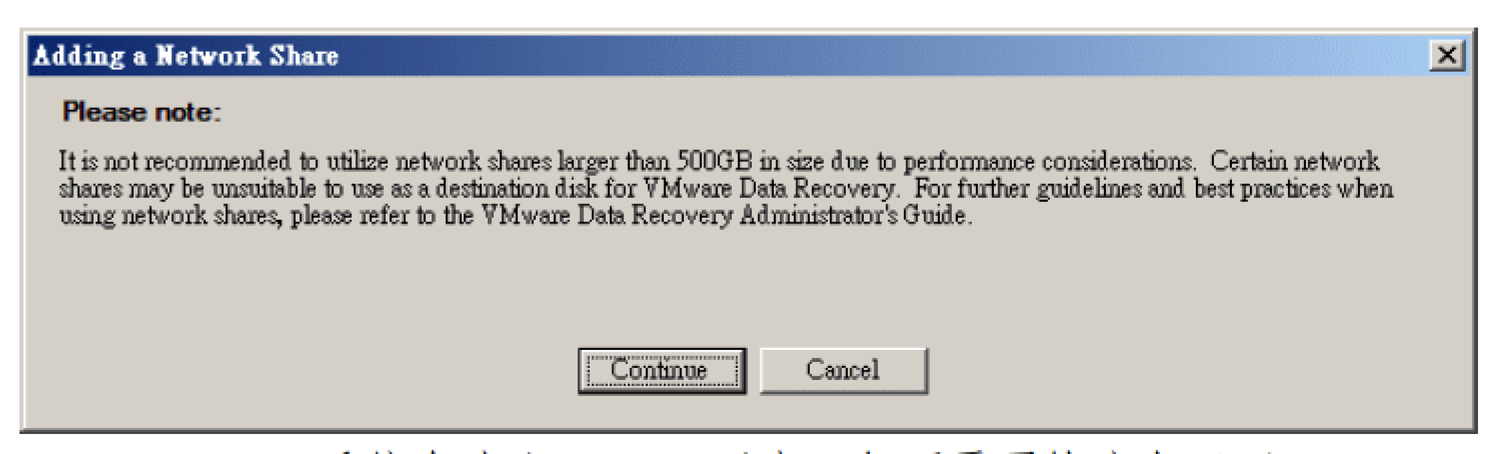
▶ 替 VDR 加入 NFS

1. 还是进入 VDR 的画面，并且选择 Add Network Share。



▲ 新建网络共享

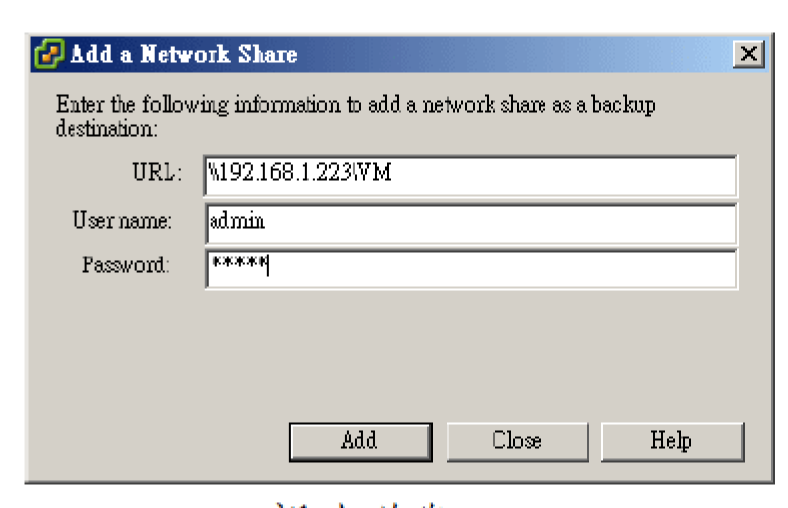
2. 此时会建议你的 NFS 有 500GB 以上空间，当然不到 500GB 也可以作为 VDR 的备份目的。单击 Continue 按钮。



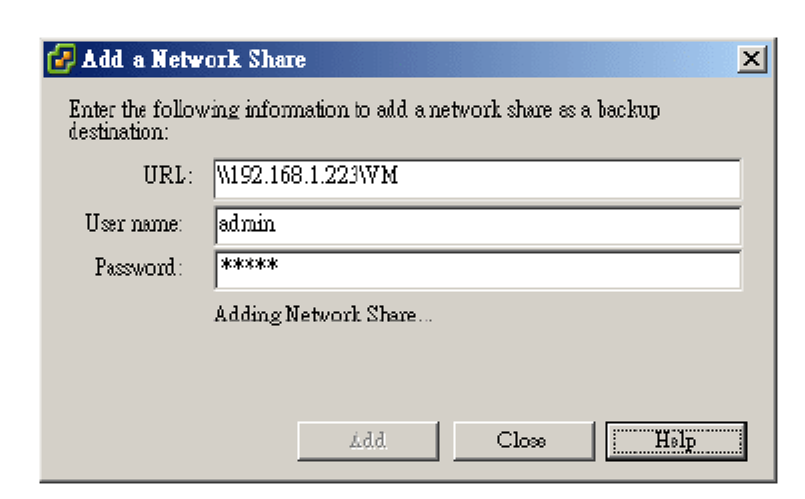
▲ 系统会建议 500GB 以上，但不需要管这个画面

3. 接下来键入 NFS 的路径及账号口令，单击 Add 按钮。注意在连接时，一定要是这个 VDR 同一个网段的存储设备。

4. 此时会开始试着连接。

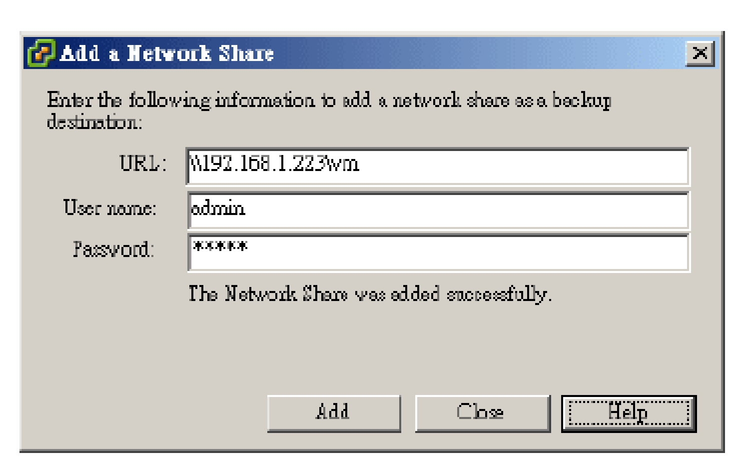


▲ 新建磁盘



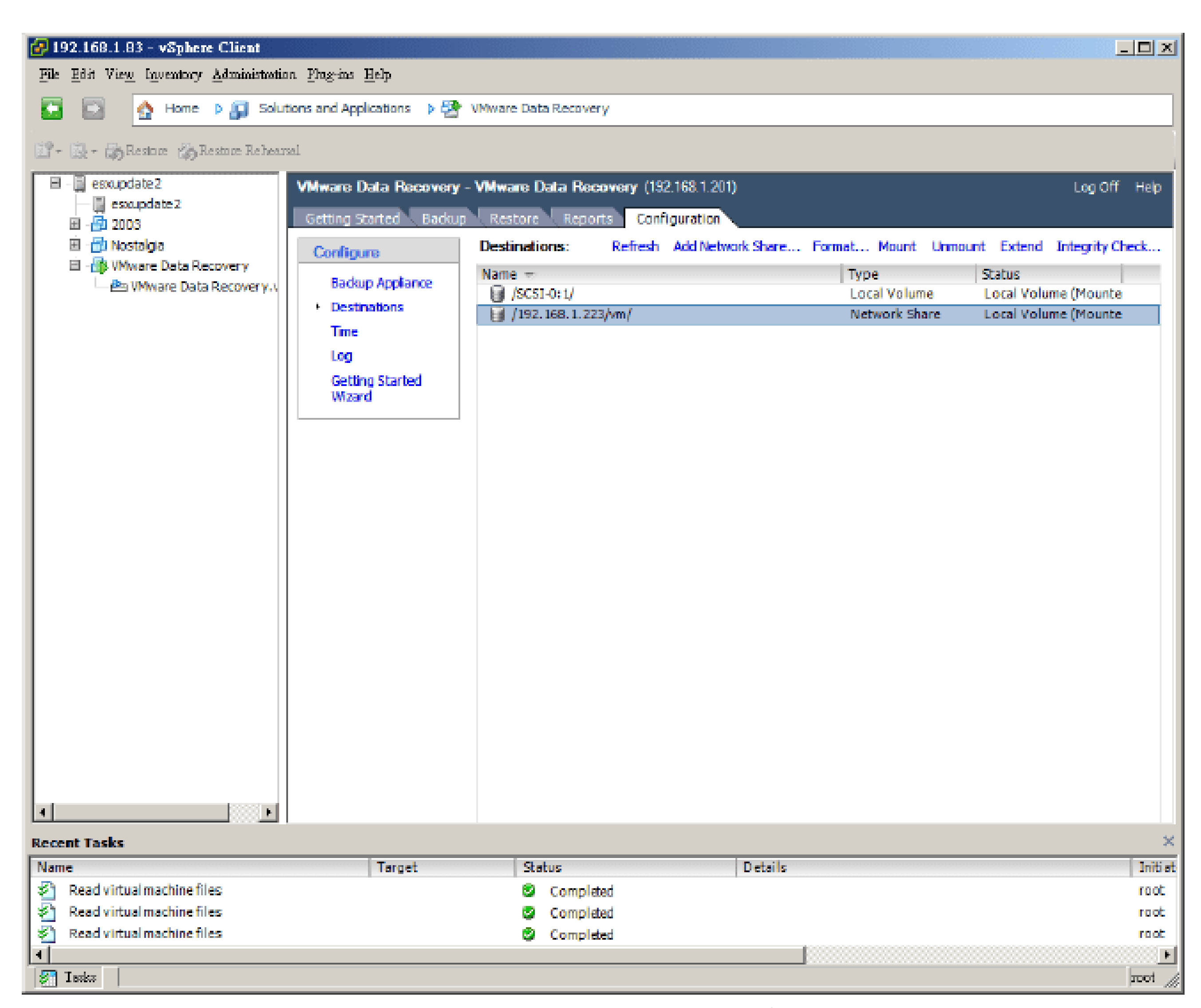
▲ 试着连接

5. 当弹出如下图所示的画面时，意味着连接成功。



▲ 连接成功

6. 我们在 VDR 的主画面中也可以看到这个设备了。



▲ 在 VDR 的主画面弹出了

21.4 使用 VDR 备份及还原 VM

当我们安装好 VA 以及配置好存储设备之后，就可以开始备份 VM 了。VDR 最大的特色就是可以在引导时备份 VM，而备份也可以配置策略来保留不同的备份。由于 VDR 的备份十分简单，我们就来看看几个最常见的应用。

21.4.1 使用 VDR 备份 VM

第一次备份就是选择 VM，并且配置备份的策略，最主要就是新建备份任务。我们就来看看

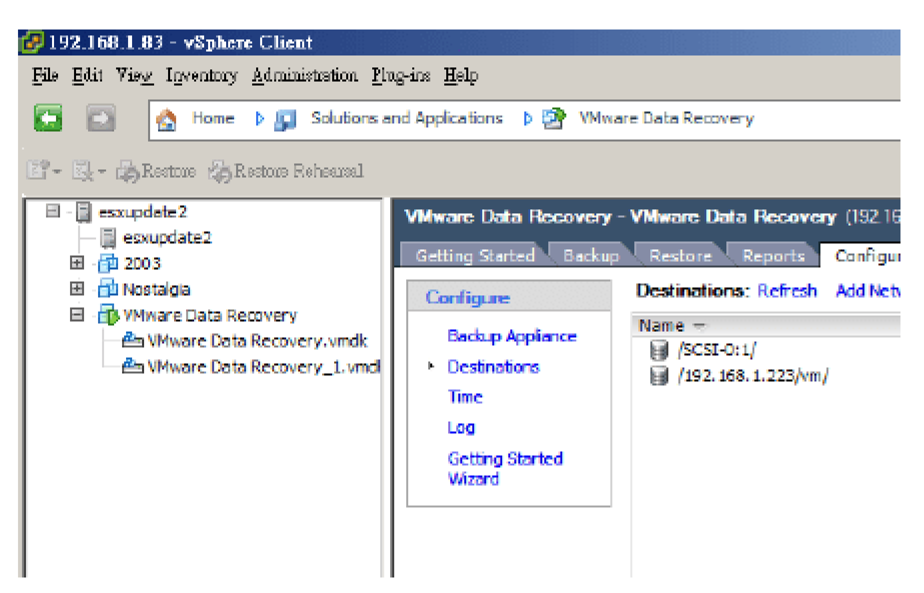
直接备份和使用向导两种方式。

1. 使用 VDR 的向导备份

使用向导可以让你在第一次备份时省下很多麻烦，我们就来看看 VDR 提供的向导使用方式。

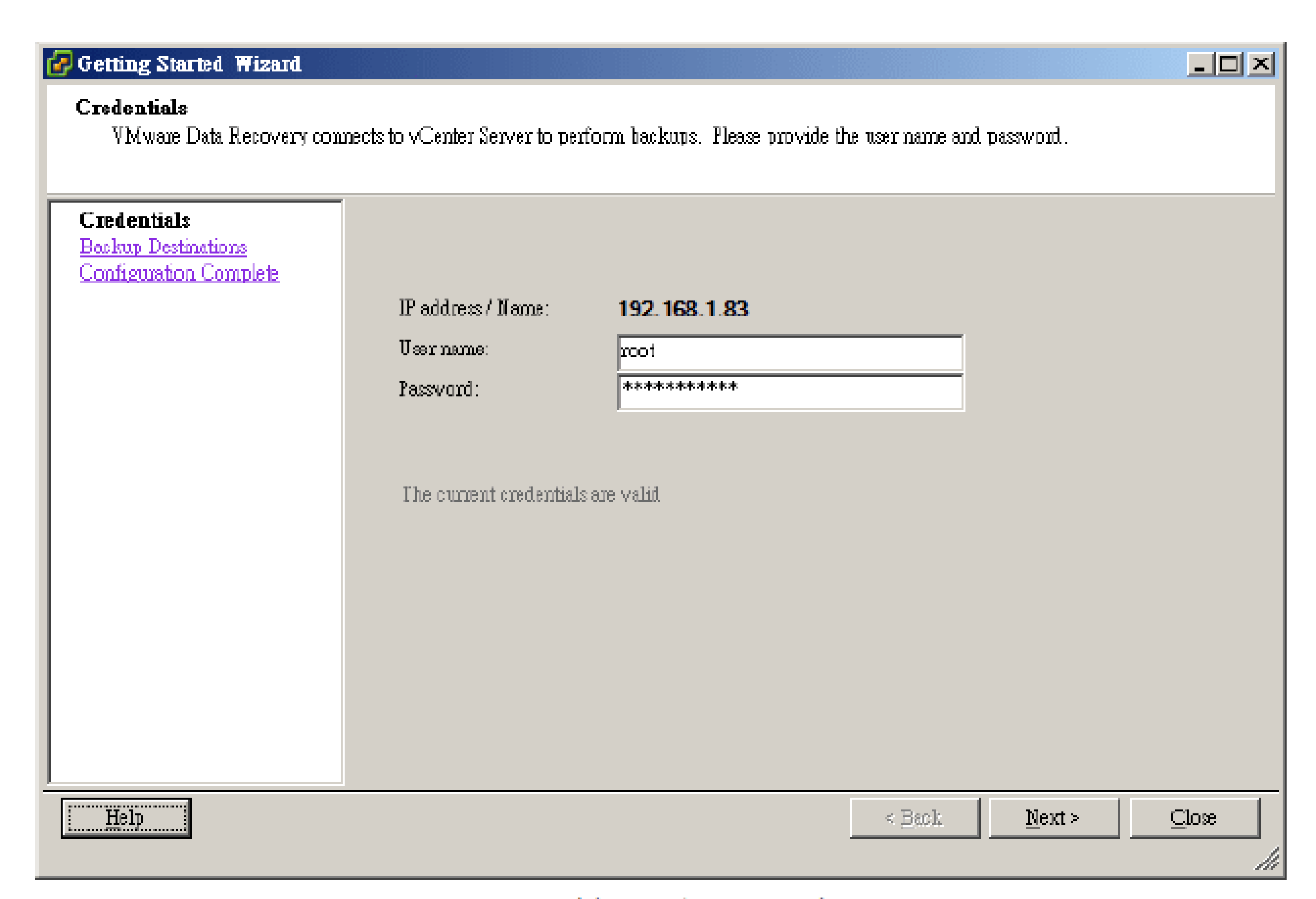
▶ 使用 VDR 的向导备份

1. 首先进入 vSphere Client 的界面，并且选择 Configuration/Getting started wizard 选项。



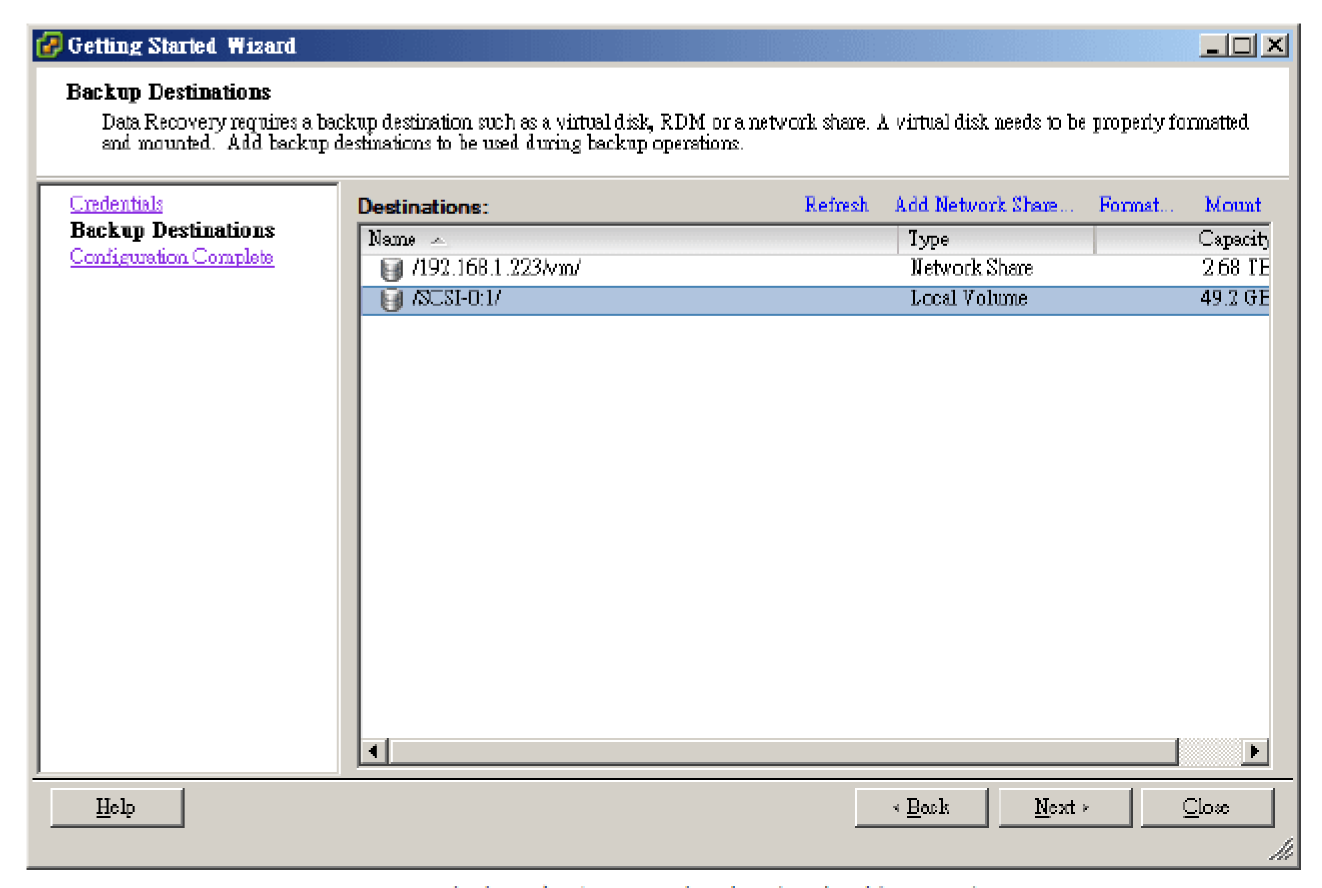
▲ 选择备份向导

2. 键入这台 ESX 的账号口令，单击 Next 按钮继续。



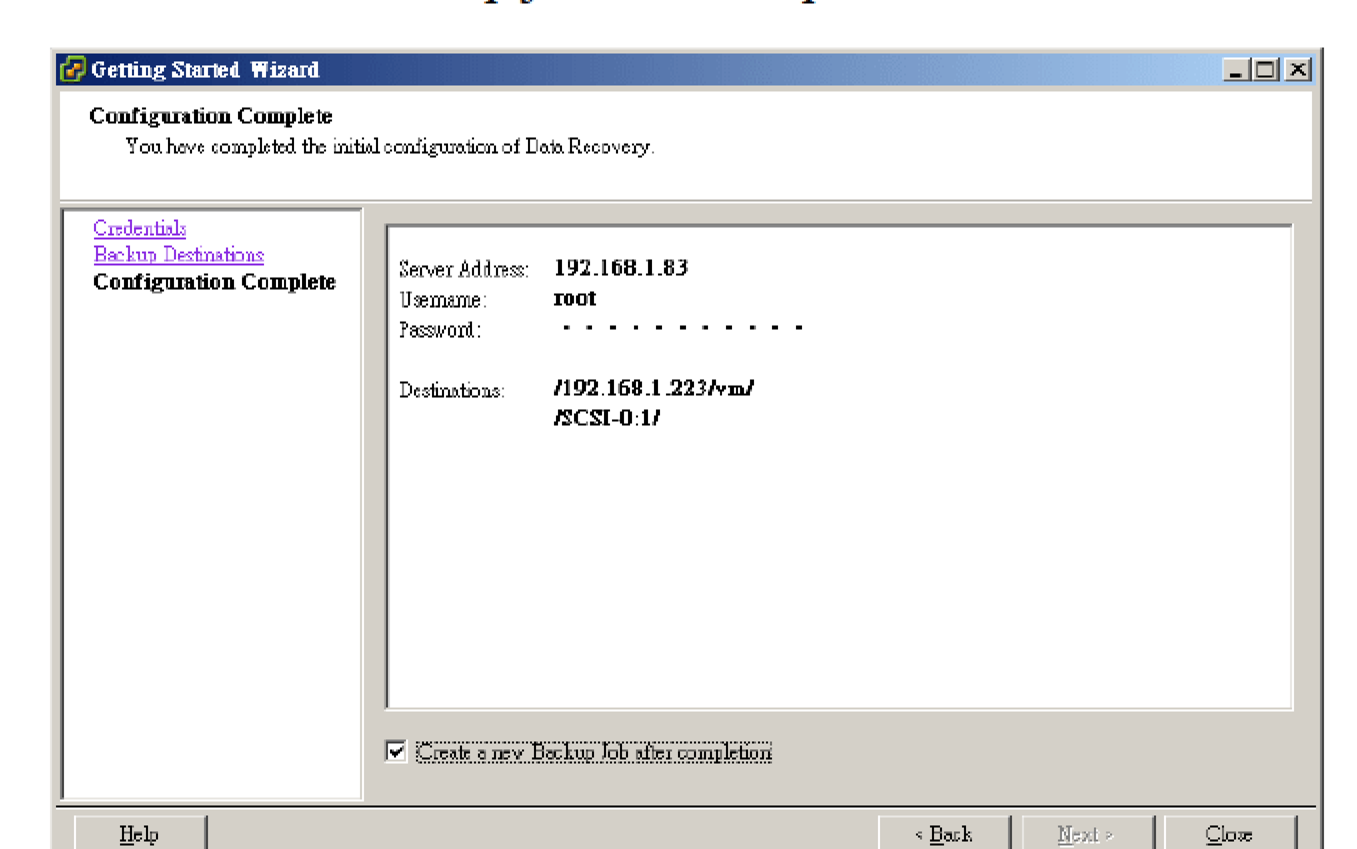
▲ 键入账号口令

3. 选择要备份的基本存储，我们就先以虚拟硬盘为主，选定后单击 Next 按钮继续。



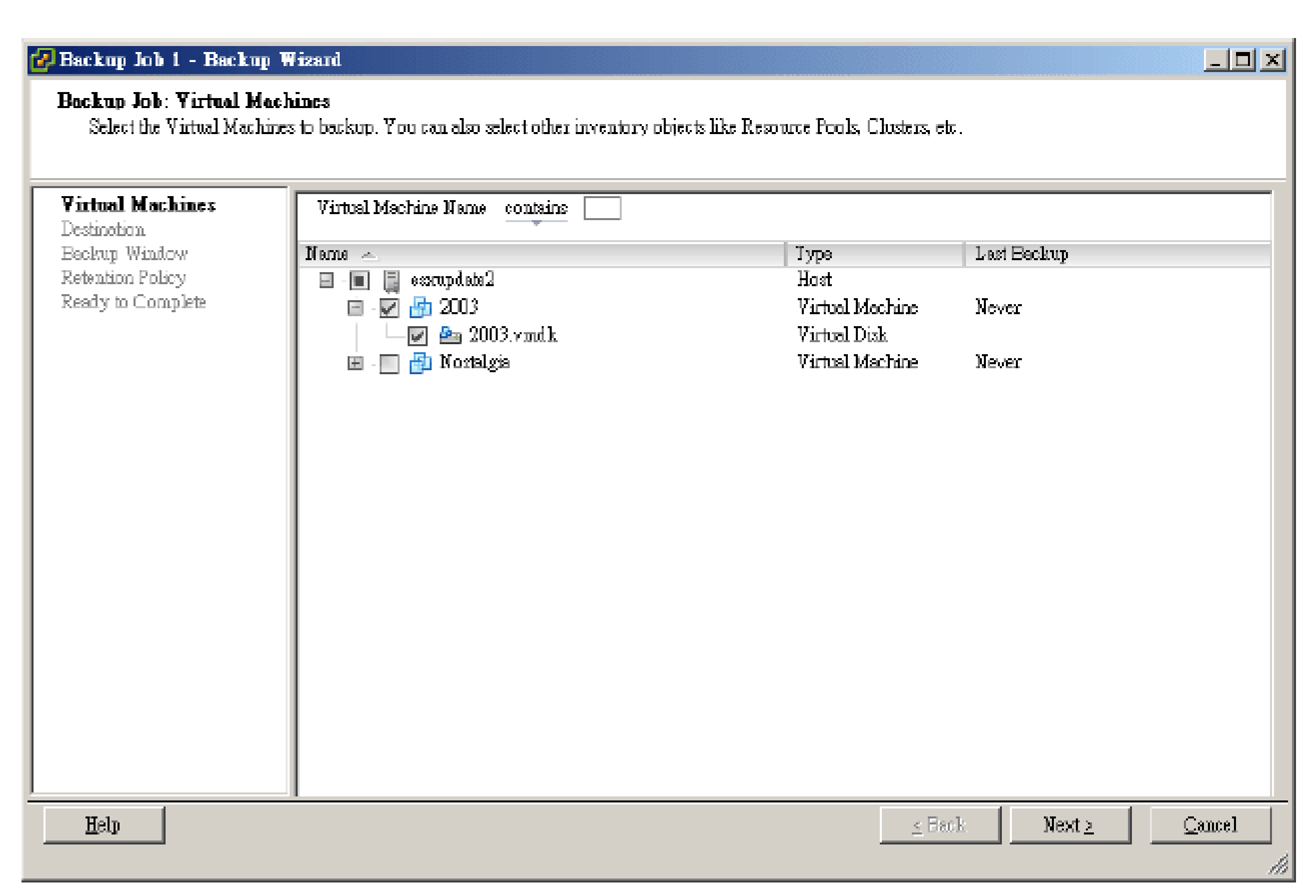
▲ 选择虚拟硬盘来当备份目的

4. 接下来将 Create a new Backup job after completion 复选框选中，并且单击 Close 按钮。



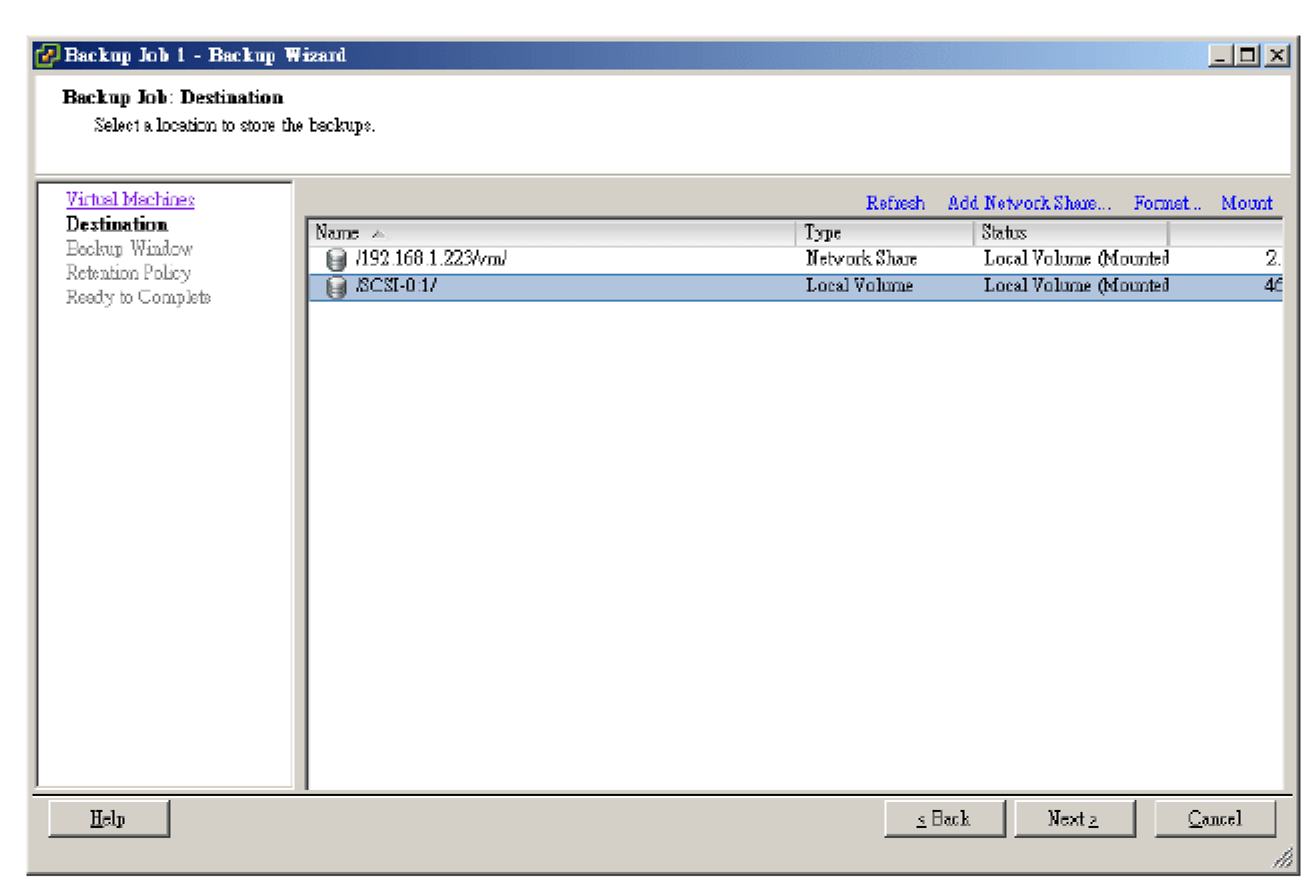
▲ 首先要创建一个备份任务

5. 接下来会弹出要备份的 VM 列表，我们就以 2003 为主。你也可以选择备份文件。单击 Next 按钮继续。



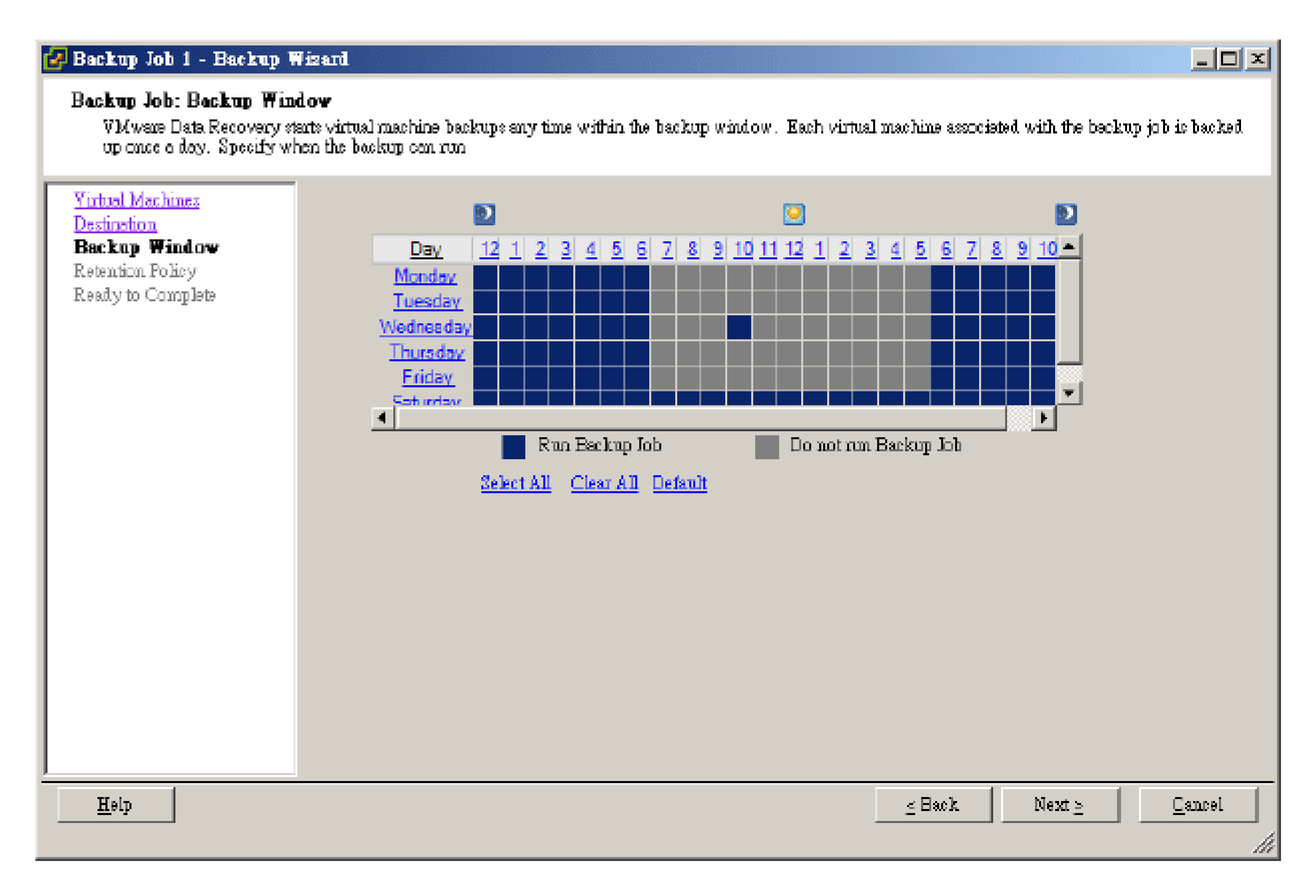
▲ 选择要备份的 VM，可以多选

6. 接下来是选择备份这个 VM 的目的地，我们就选择本地硬盘。单击 Next 按钮继续。



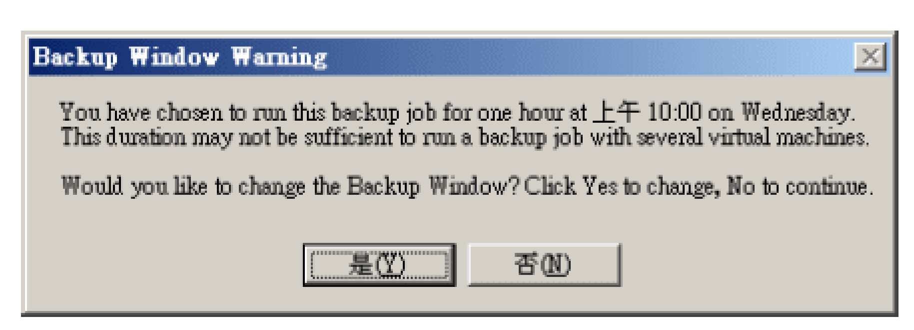
▲ 要重新键入一次备份目的

7. 接下来是选择备份这个 VM 的时间。VDR 会先做一次全备份，接下来的时间都会做增量备份。这里是选择每小时的备份点。你会发现在周一到周五的上班时间，VDR 是不建议备份的，当然你也可以直接单击，弹出深色就意味着这个时间点要备份。单击 Next 按钮继续。



▲ 这里是备份的时间点

8. 如果你有选择上班时间，VDR 会再次询问你。单击“否”按钮才会继续。

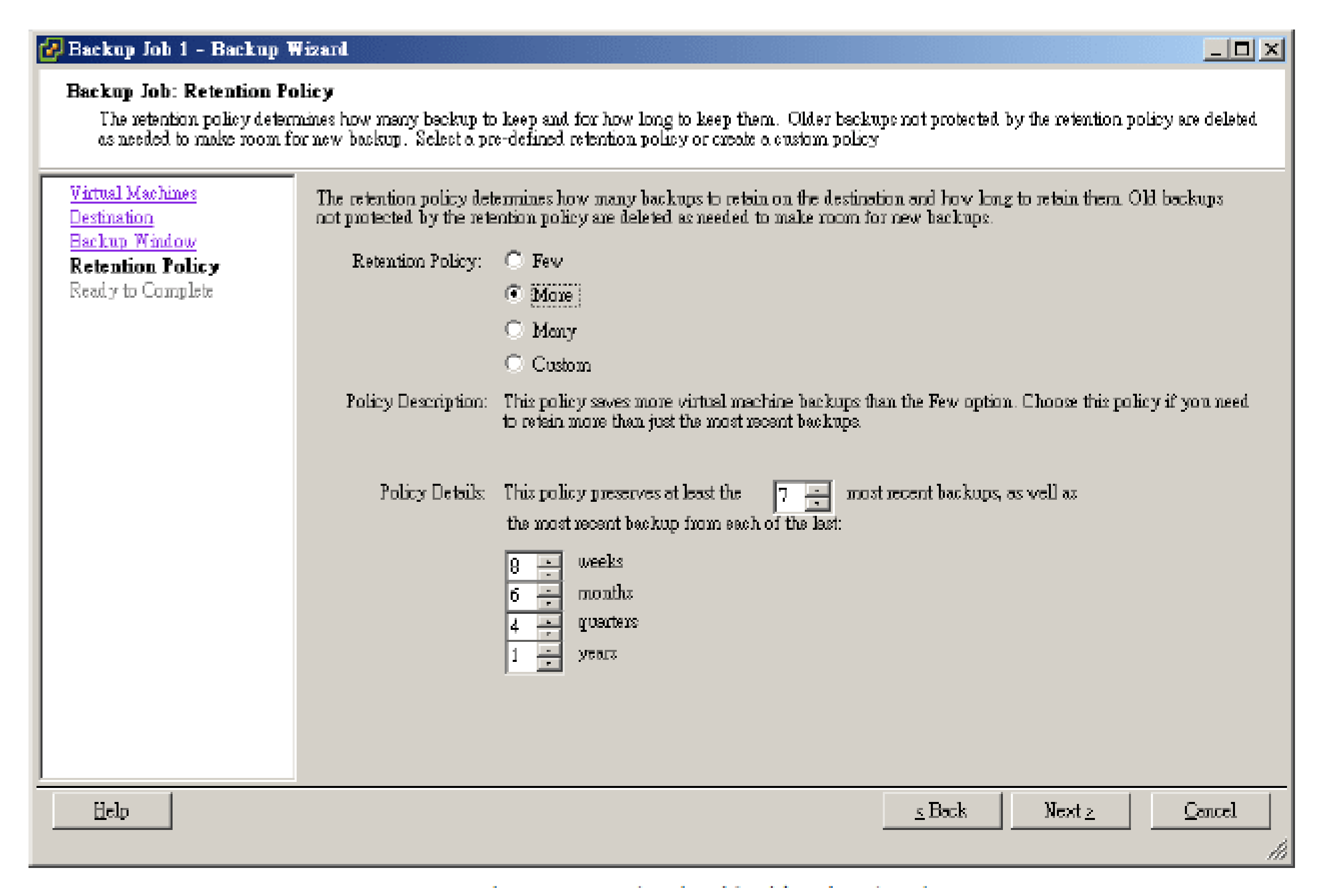


▲ VDR 不建议在任务时间备份 VM

9. 接下来是备份策略的选择。VDR 会保量多个备份时间点，换句话说，你可以拥有多个还原点。但由于太多备份点会浪费硬盘空间，因此 VDR 可以选择少、多一点、多、自定义。一般来说，选择 More 就够了。

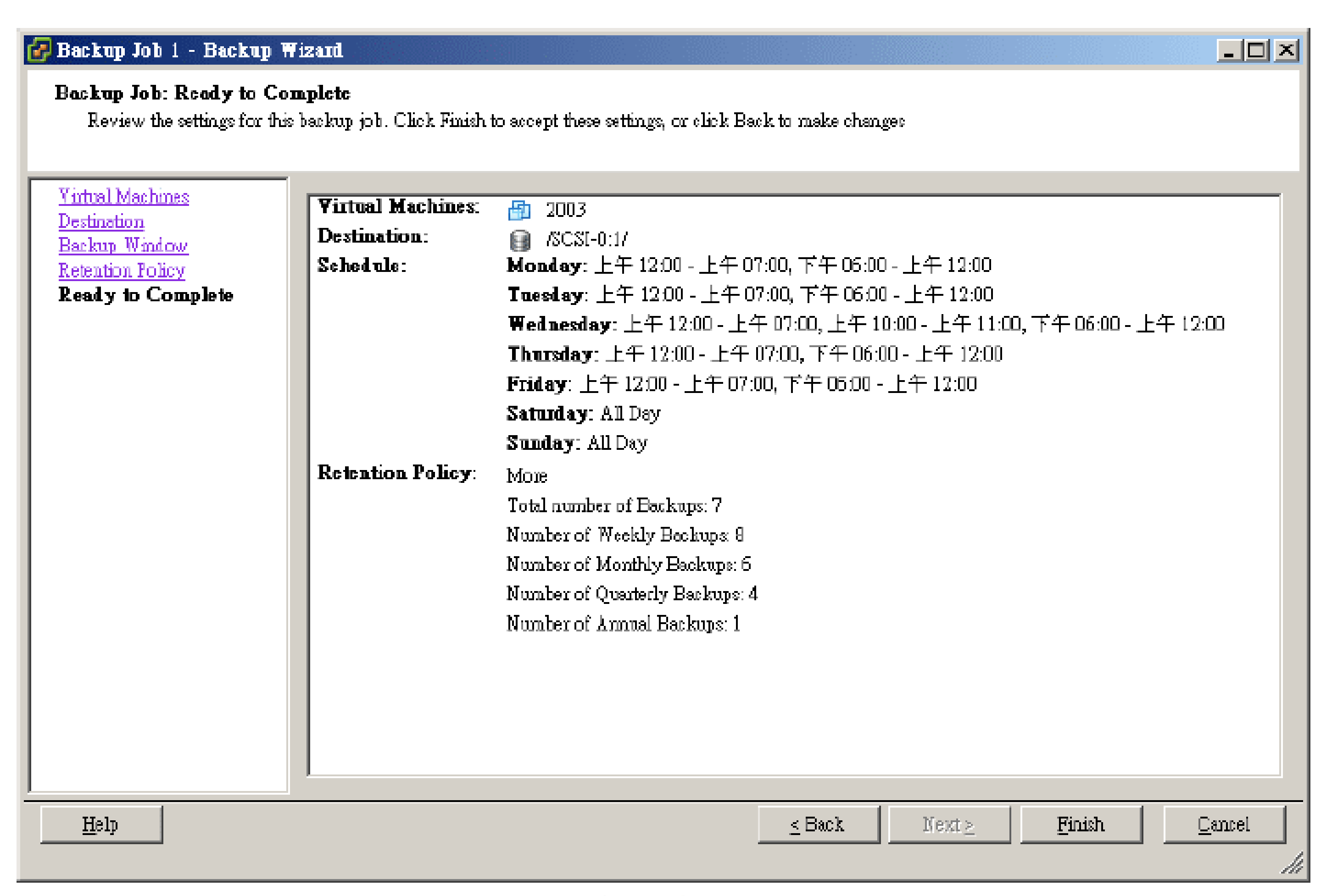
10. 接下来是保留的备份点。系统默认值为 7。

11. 另外下面的时间点，则是保留备份的时间点。如 8 周内、6 个月内、4 个季度内及 1 年内等。单击 Next 按钮继续。

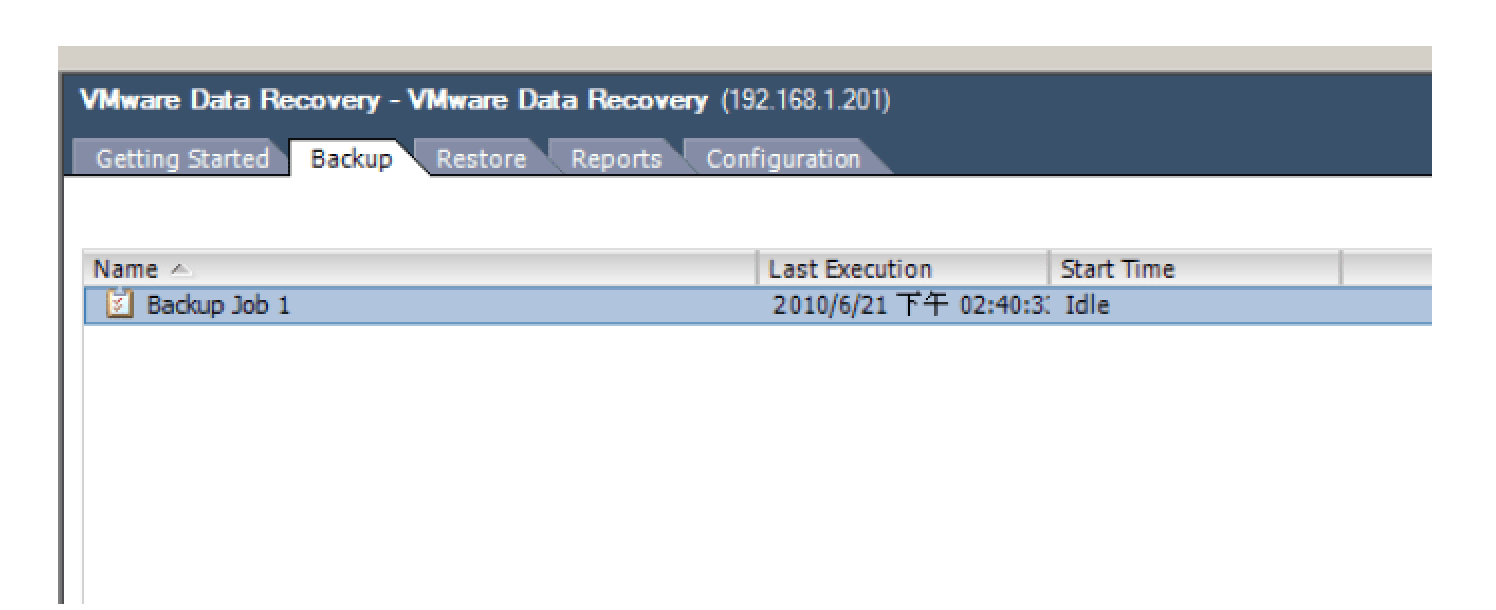


▲ 以上三项为备份策略的选项

12. 接下来是总结画面，单击 Finish 按钮则落实这个备份任务，但还没开始备份。会到之前选定的时间点开始才会开始备份。



▲ 落实备份任务



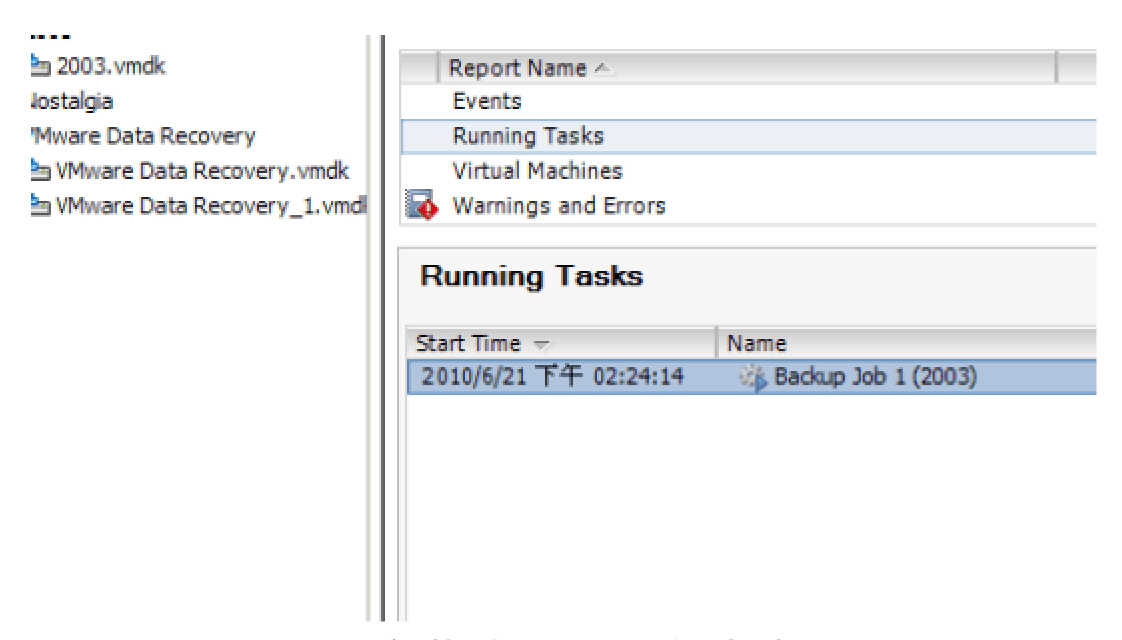
▲ 但由于是有时程安排，因此还不会开始备份，是处于 Idle 状态

2. 直接备份 VM

直接备份 VM 十分简单，在第一次创建好备份任务之后，如果时间还没到，就可以直接备份 VM，我们就来看看。

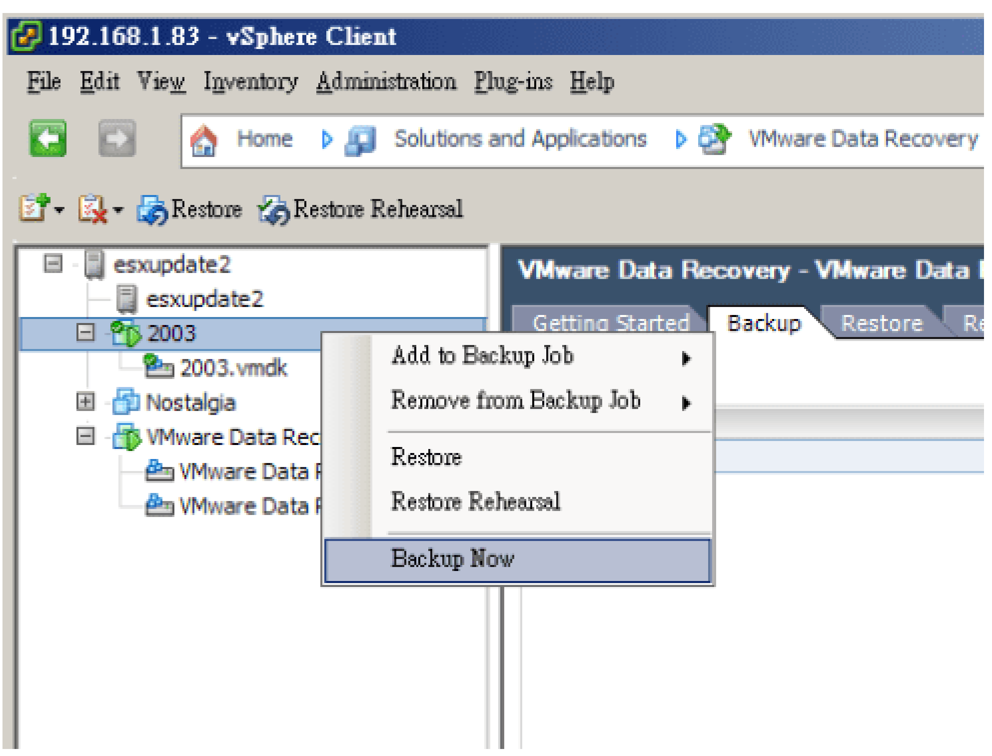
▶ 直接备份 VM

1. 确定你已经有一个备份任务，并且要备份的 VM 也在该任务中。



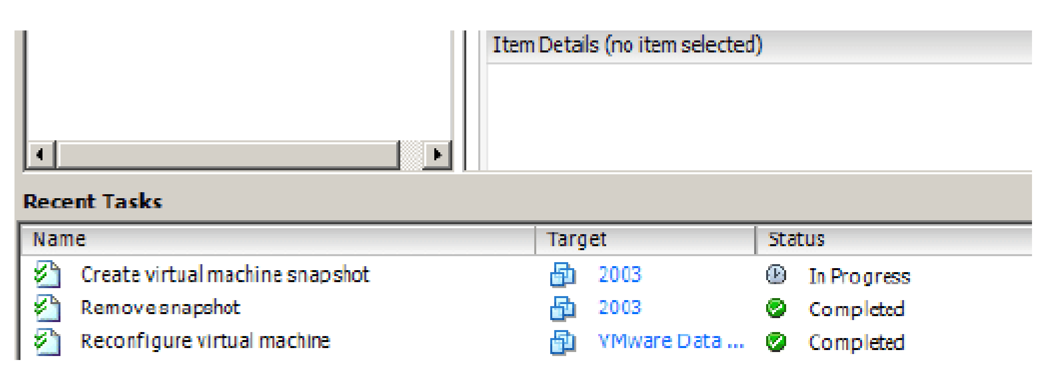
▲ 备份的 VM 已经存在了

2. 选择该任务的 VM，并且在右键菜单中选择 Backup Now 选项。



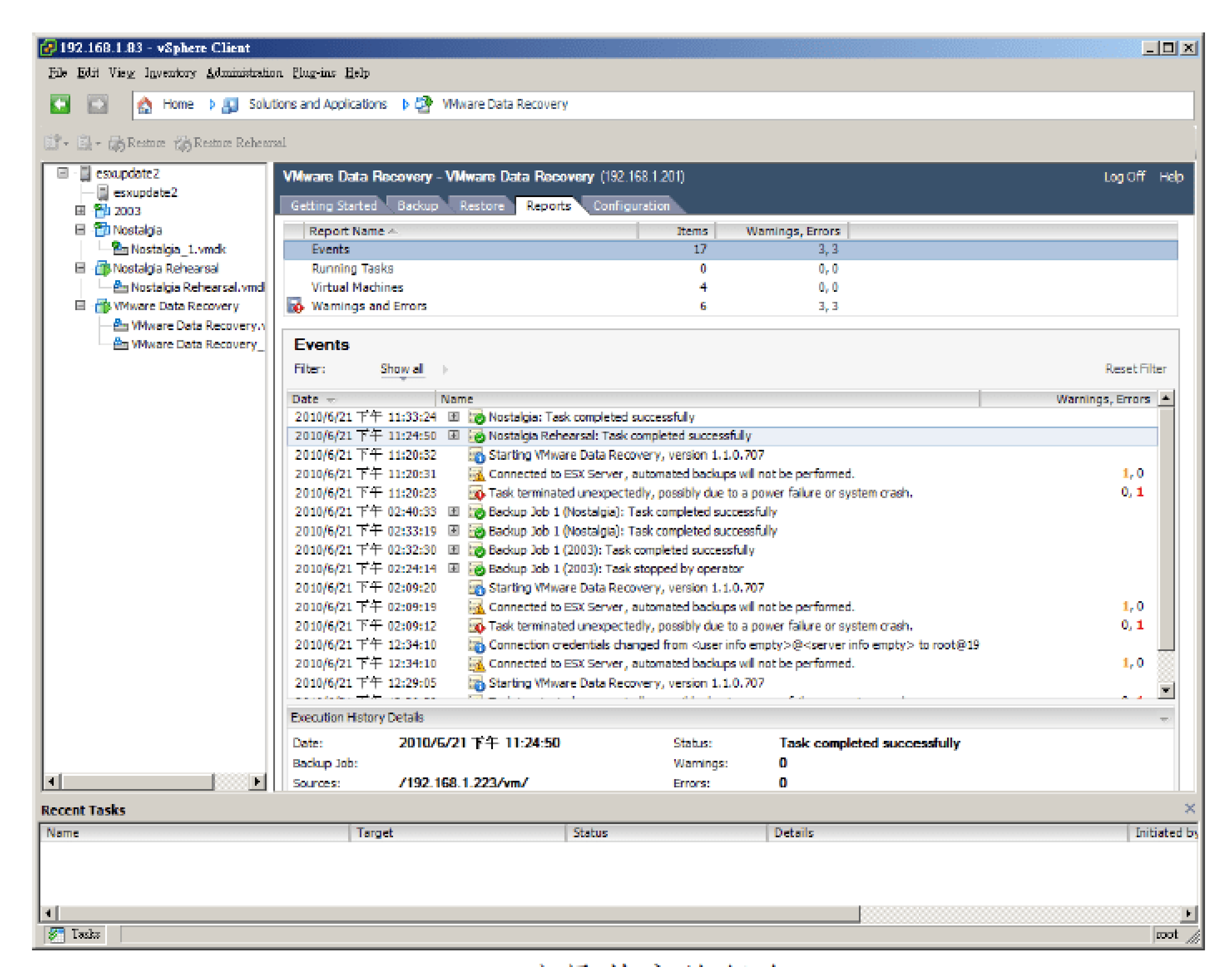
▲ 选择 Backup Now 选项

3. 此时会开始创建这个 VM 的快照，并且准备开始备份任务了。



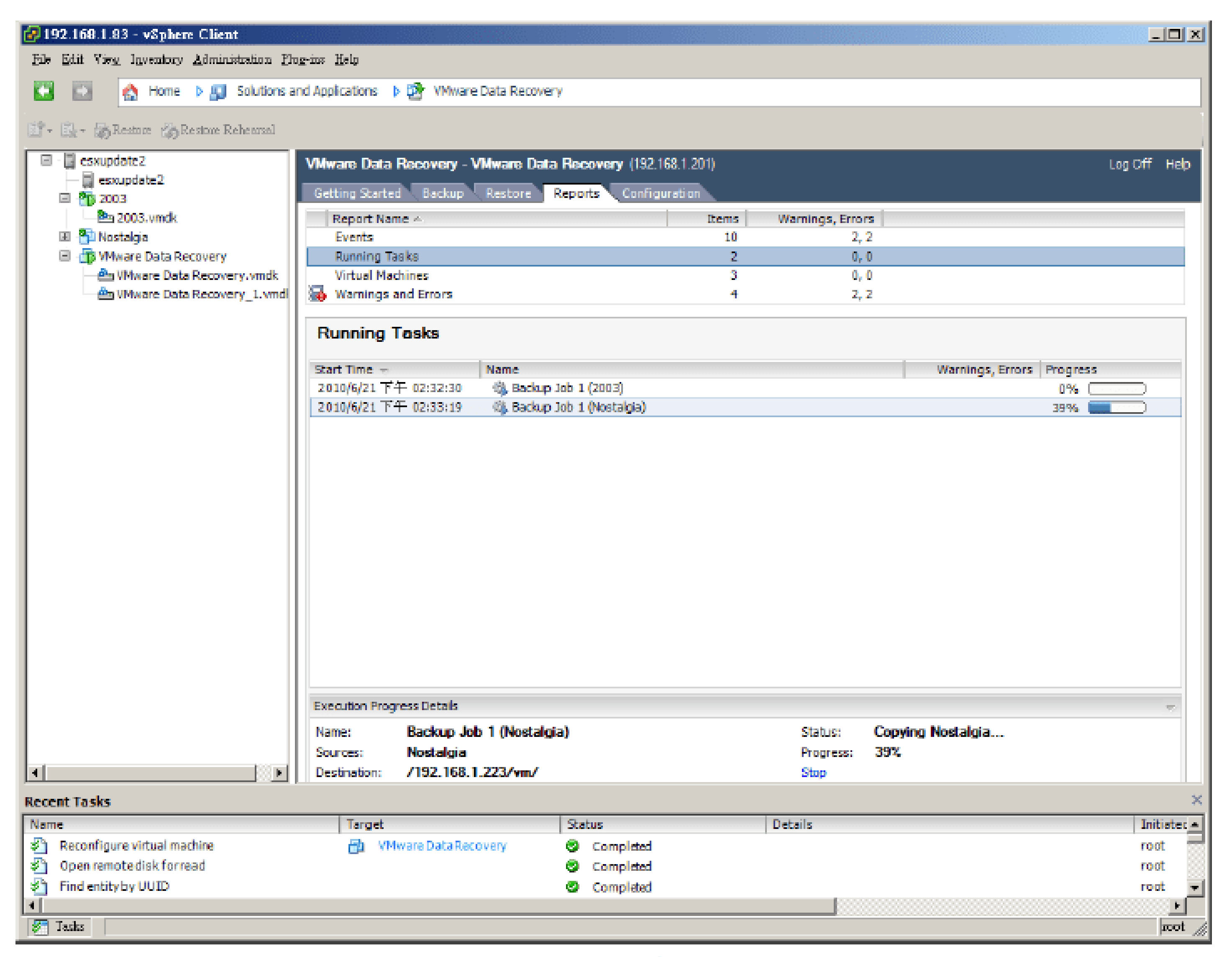
▲ 先创建快照

4. 我们可以到 Report 来检视备份的任务，如 Events 会给出落实的任务报告。



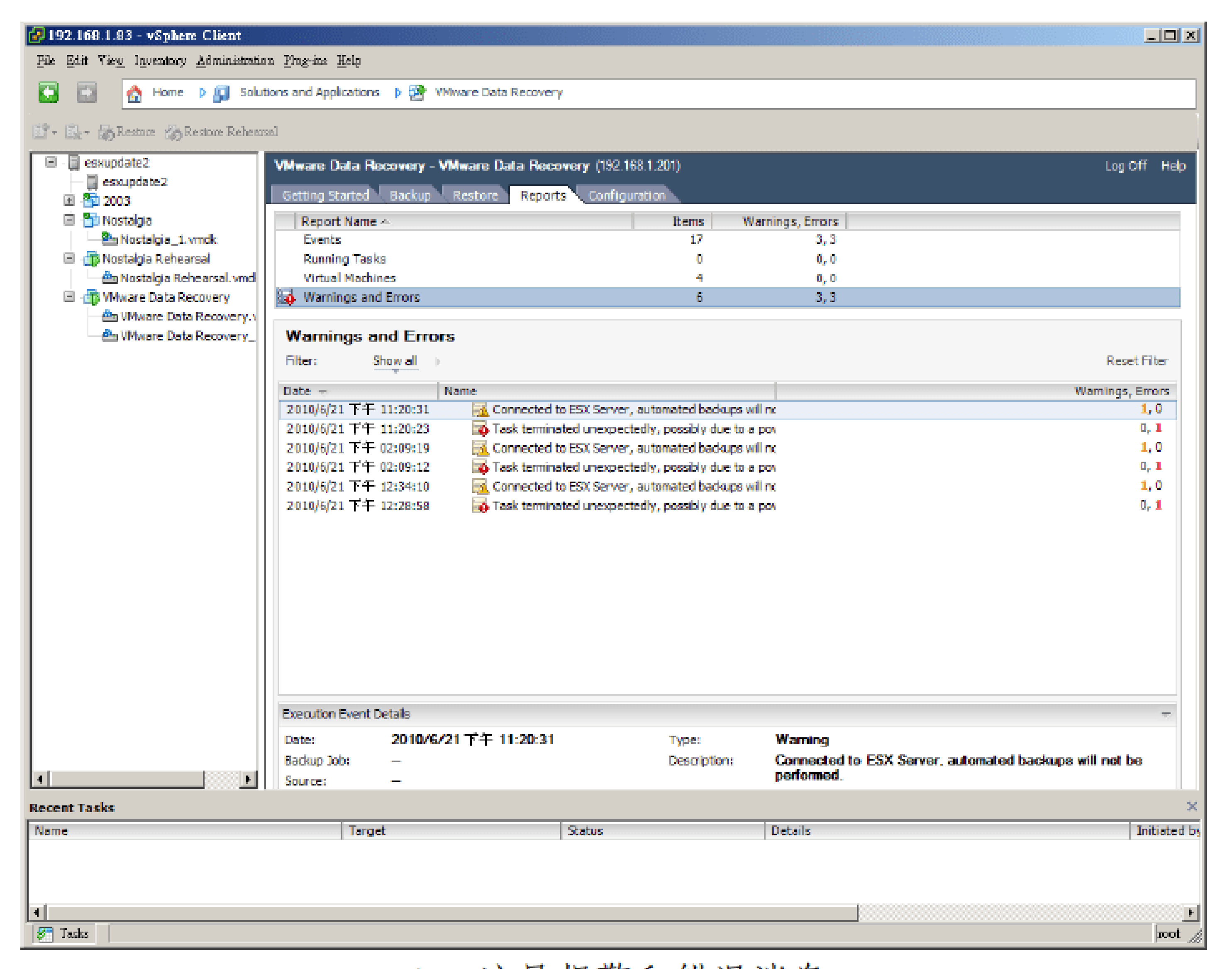
▲ 这是落实的任务

5. 在 Running Tasks 中可以看到当前进行的任务。



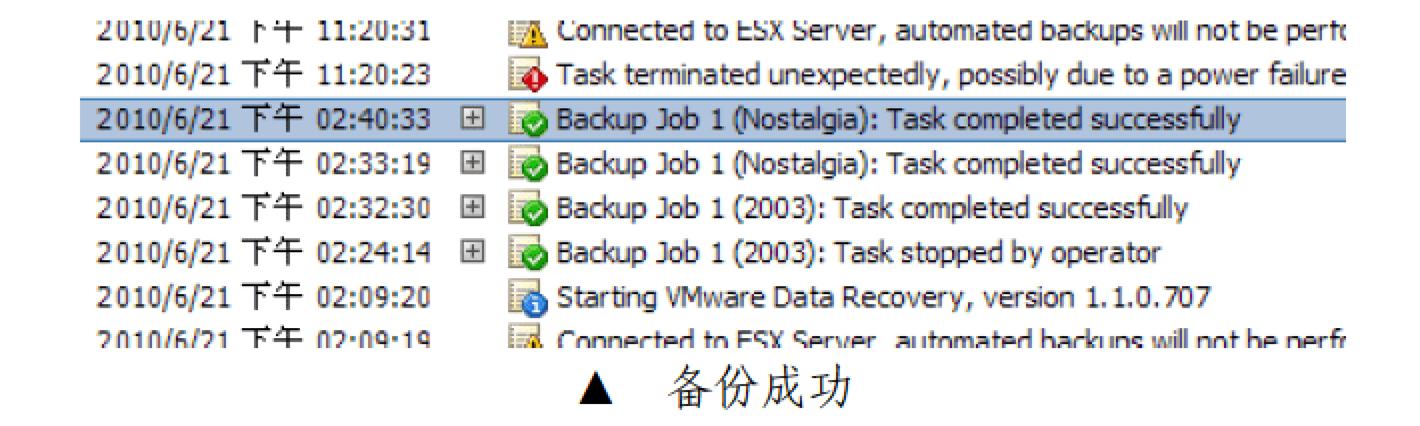
▲ 这是当前的任务

6. 我们也可以看到 Warnings and errors 弹出的报警消息或错误。



▲ 这是报警和错误消息

7. 当备份落实后，会自动在 Events 中留下表项。



▲ 备份成功

21.4.2 使用 VDR 还原 VM

使用 VDR 还原 VM 是非常方便的任务，你可以选择任何一个还原点来还原。如此一来，并不一定是最新的图像被还原。举例来说，如果你的 VM 在某个时间点的服务常规，但之后发生问题，最后导致整个 VM 不能用，那么在服务常规之后的还原点虽然对比新，但却是有问题的备份，因此你就可以选择较之前的时间点来还原，让 VM 的服务回到常规的那个时候。

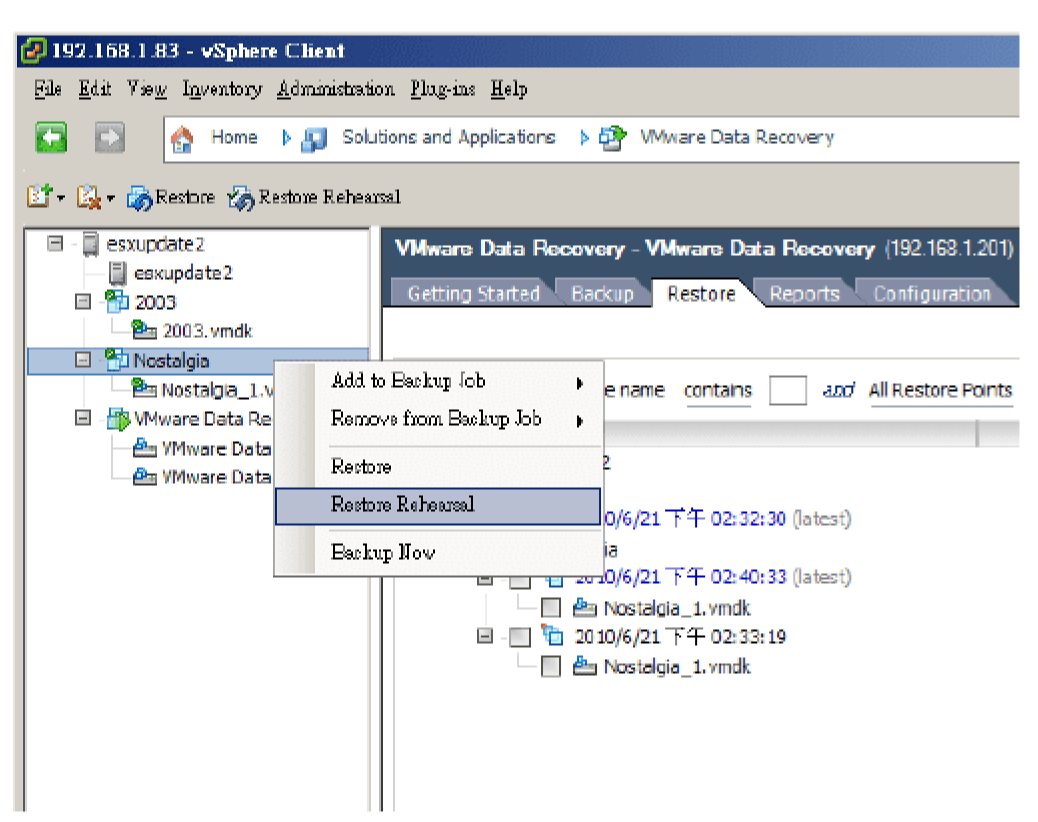
由于大部分的还原问题都较备份来的多，因此 VDR 也提供了一个“还原排练”（Restore rehearsal）功能，让你在真正还原之前先演练一次，我们就来看看演练的还原任务，再来看完整的还原做法。

1. 还原演练

还原演练就是“假装”还原 VM，测试整个还原过程有没有问题，我们就来看看还原演练的实践。

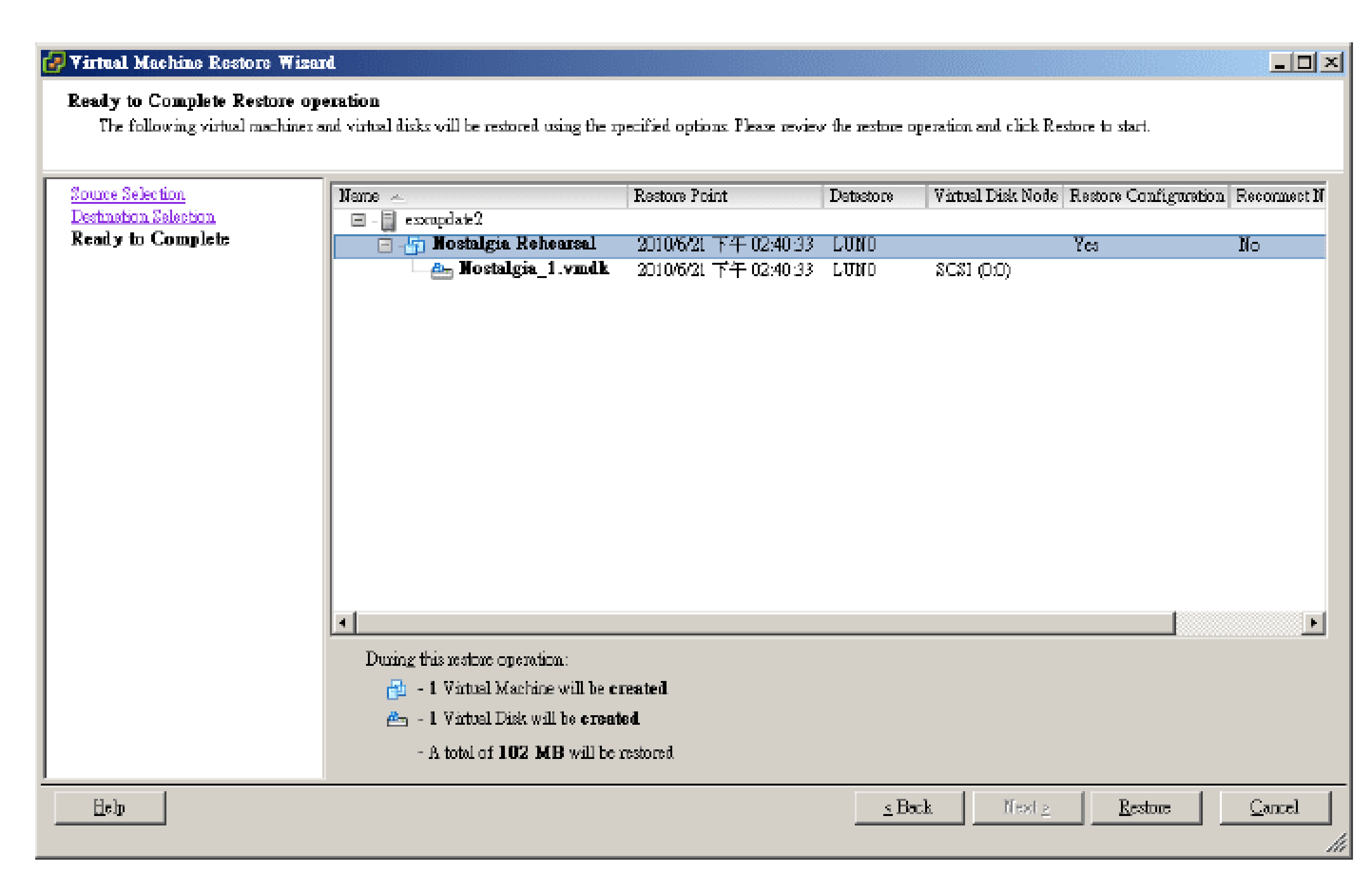
▶ 使用 VDR 还原演练

1. 选择要还原演练的 VM，前提是该 VM 已经有备份数据存在。在其上右击，选择 Restore Rehearsal 选项。



▲ 选择还原演练

2. 此时会弹出该 VM 的状态，直接单击 Restore 按钮即可。



▲ 开始演练

3. 接下来我们可以看到还原开始，当 100%时就还原完毕。

▲ 还原开始了

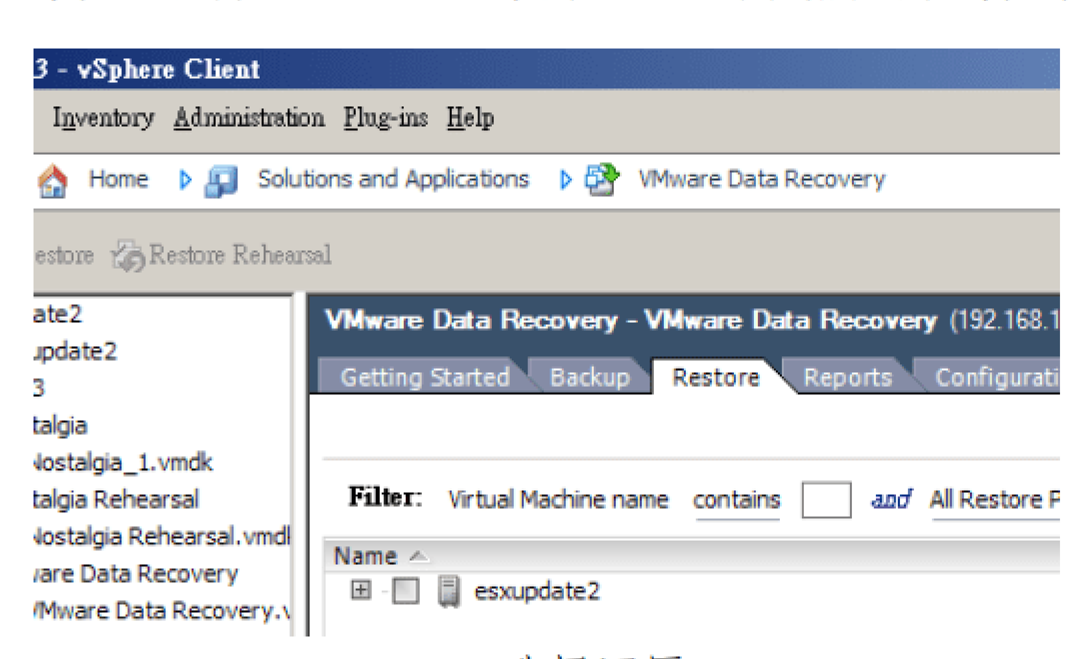
也可以在 Events 中看到还原演练成功落实。

2. 真正开始还原

还原 VM 最大的问题就是现有 VM 将会被覆盖。因此 VDR 会提出报警。但只要你确定还原的 VM 是可以覆盖的，就直接将该 VM 还原回去即可。下面就是步骤。

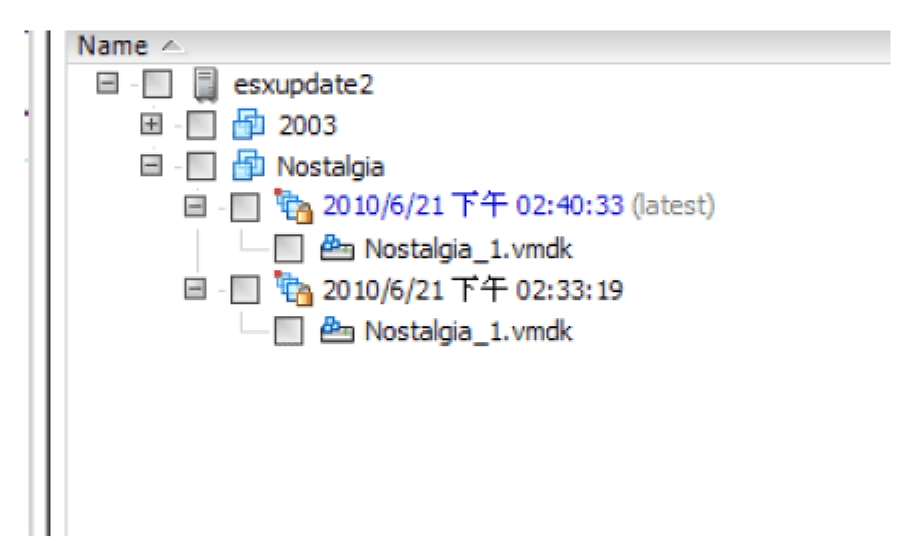
▶ 使用 VDR 还原 VM

1. 进入 VDR 中，并且选择 Restore 选项卡。此时我们可以看到已经有备份 VM。



▲ 选择还原

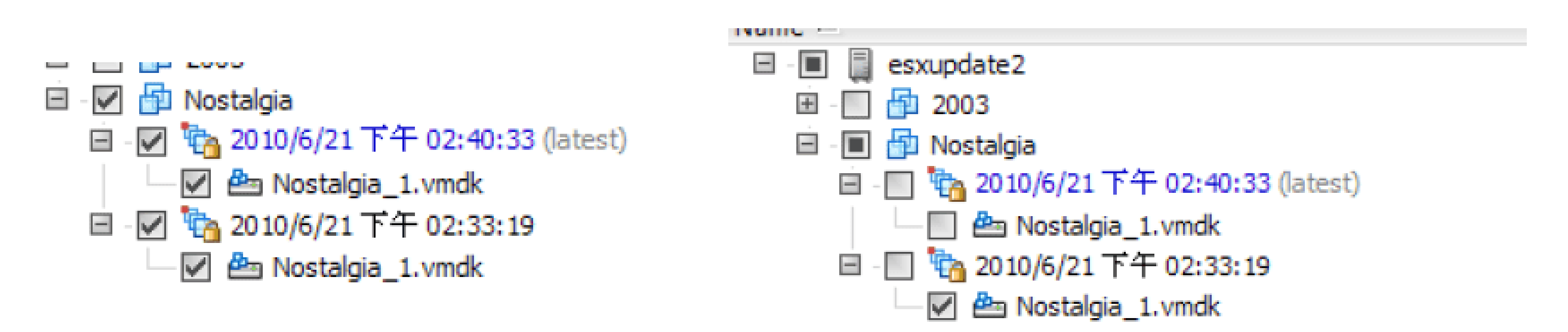
2. 将你要还原的 VM 打开，你会发现有多个还原点，系统会给出最新的还原点。



▲ 通常一个 VM 会有多个还原点

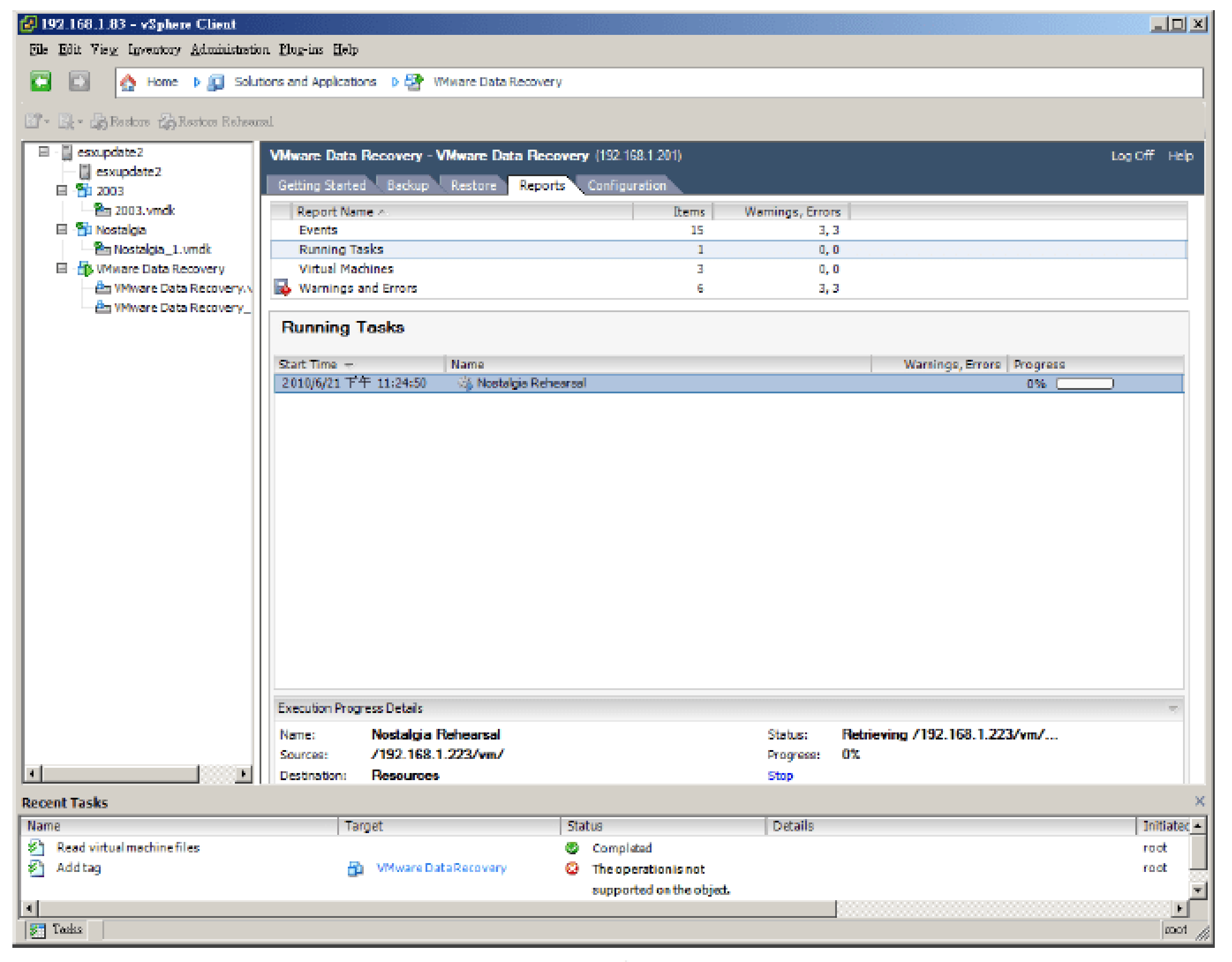
3. 如果你想要做完整的还原，可以直接选取整个 VM，系统会将整个系统还原，到最新的时间点。

4. 如果你想还原较旧的时间点，可以只选择某一个 VM 还原点。



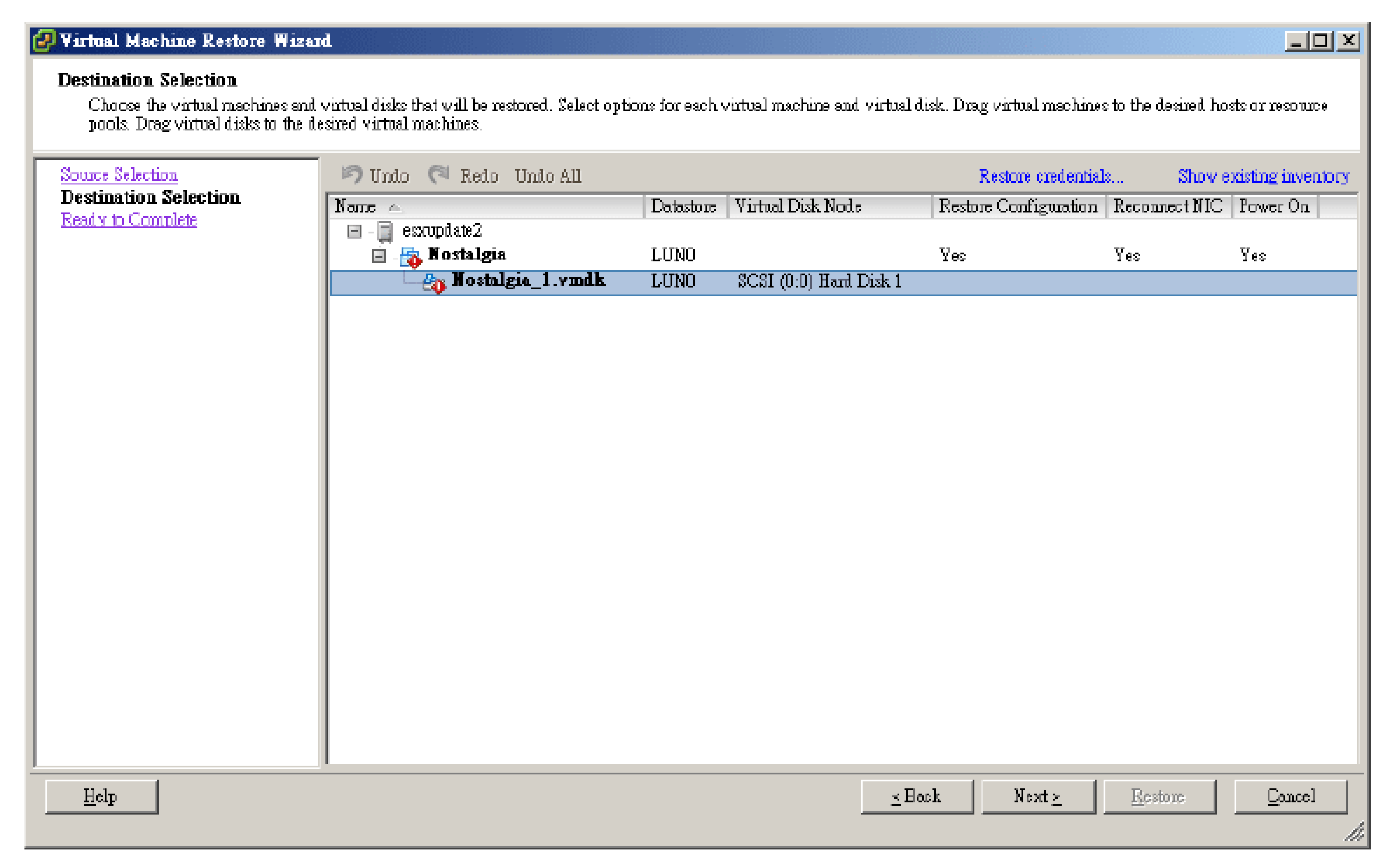
▲ 选取整个 VM 会做最新还原点的还原　　▲ 可以只还原某个时间点

5. 选择完之后，直接单击 Restore 按钮



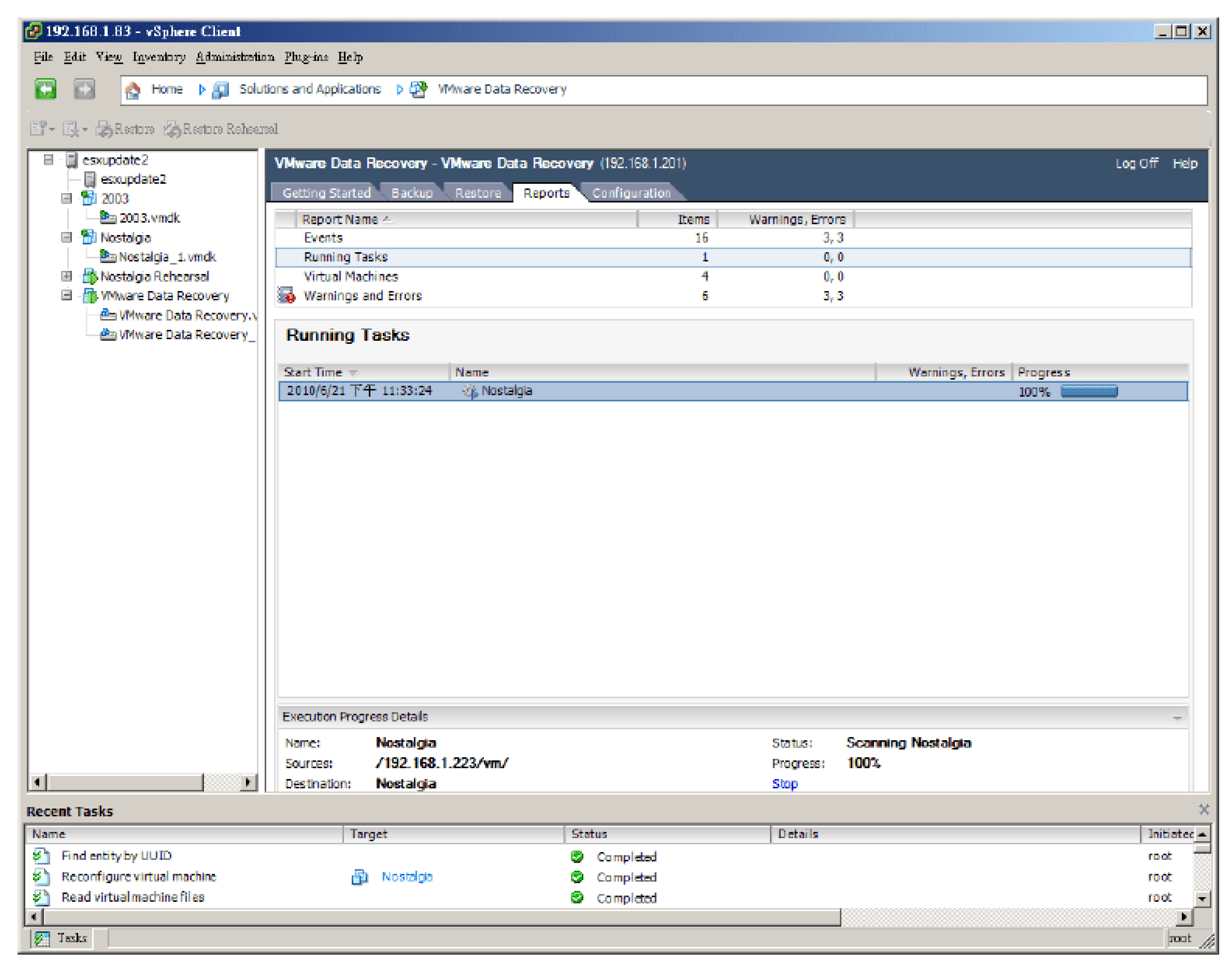
▲ 开始还原

6. 此时系统会弹出要还原的 VM，单击 Next 按钮继续。



▲ 弹出要还原的 VM

7. 此时会开始还原，当还原到 100%时，还原即落实。



▲ 还原落实

也可以在 Events 中看到还原落实以及错误或报警。

结 语

从 vSphere 开始，大部分的高级功能都以 VA 的方式提供，最典型的就是 VDR。VMware 是 OVF 的创始者之一，将大部分的 VA 用 OVF 格式来提供，不但有更强的兼容性，也方便了软件厂家交付新服务。当然 VDR 就是 VMware 本身最好的服务，笔者也十分推荐使用 VDR 取代 VCB，尤其是在较大型的企业环境中。而使用了 VDR 之后，VM 的安全性在 HA、DRS 及 VMotion 之后，更能获取进阶的保障。

读书笔记

第 22 章 理解 vSphere 的 vNetwork Distributed Switch

关键词：

- 什么是 vNetwork Distributed Switch
- vDS 的特色有什么
- 为什么要使用 vDS
- vDS 的基本部件
- 如何创建 vDS
- 什么是 DVUplink
- 什么是 dvPortGroup
- 什么是物理网卡
- 什么是虚拟网卡
- 如何创建及操作物理网卡
- 如何创建及操作虚拟网卡
- 如何配置 PVLAN

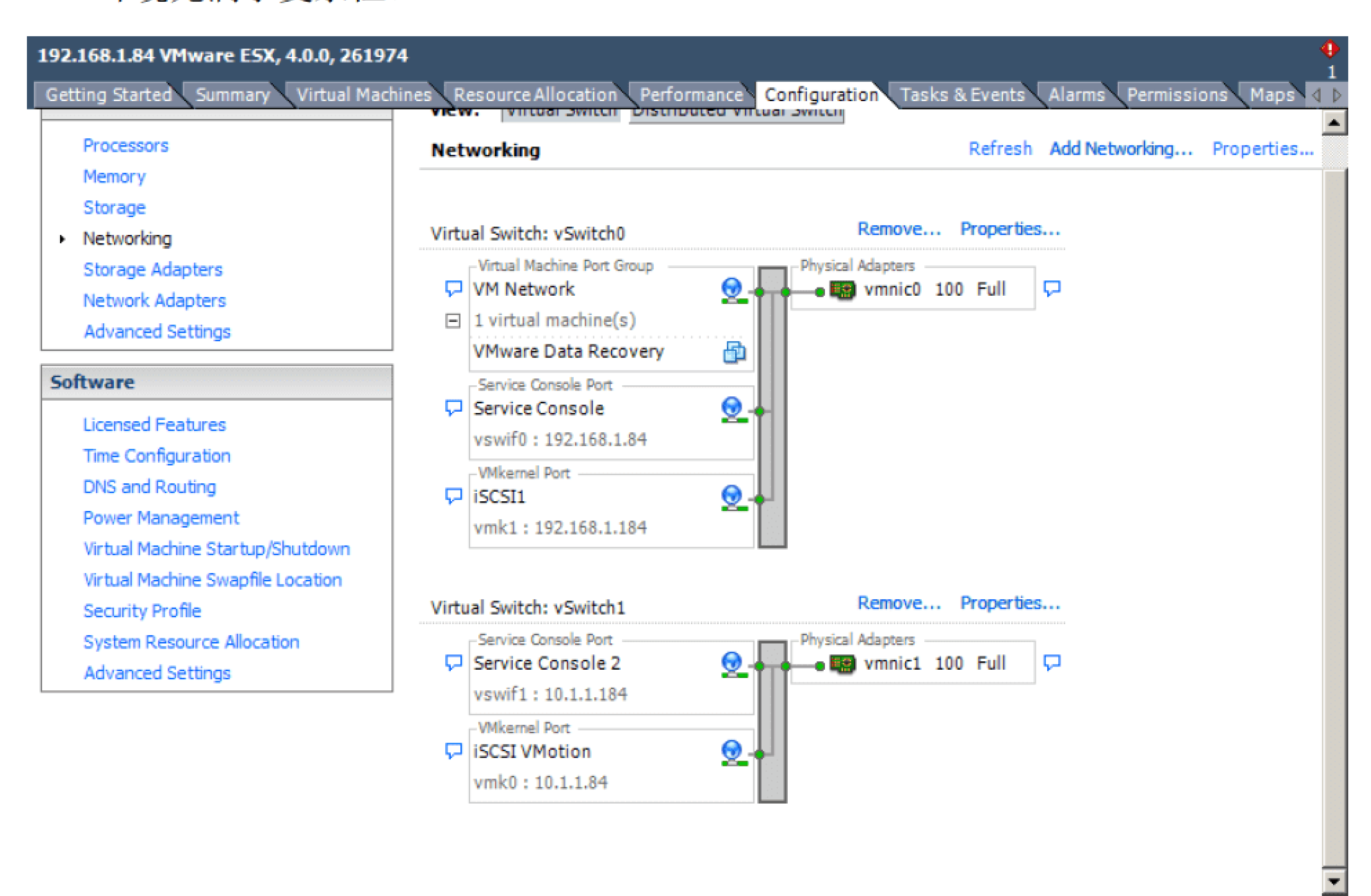
你一定很少听到什么是 Network VMotion 吧！vSphere 的推出改变了虚拟机网络任务的方式，在旧版的 VI3 中，不管是用户或是 VMware 厂家，仍然在摸索着最棒的网络管理平台，而随着第三方厂家的加入，再加上虚拟化技术的普及，VMware 在 vSphere 中加入了真正管理多台 ESX/ESXi 的单个交换部件，不但让虚拟平台和真实平台的距离更近了一步，更让虚拟化的弹性及功能性大大超越了物理环境，我们在这一章就来看看。

22.1 理解什么是 vNetwork Distributed Switch

vSphere 推出了新的网络管理方式，将原来分布在每一台 ESX/ESXi 主机上的交换机集成，变成一个单个的管理界面。许多原来 VI3 的用户虽然安装了 vSphere，但仍然使用旧式的单机交换方式管理网络，因此在这一章，我们就来看看 vSphere 中得奖无数的产品 vNetwork Distributed Switch（以下简称 vDS）。

22.1.1 网络环境的交换机

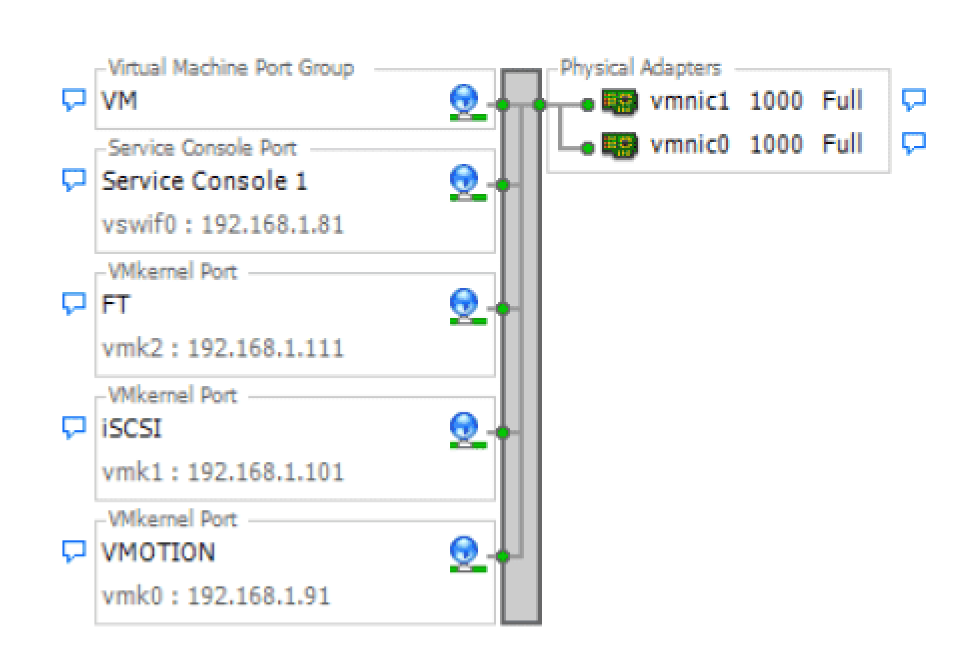
在充满了服务器的环境中，交换机的角色十分重要。交换机通常在 MAC 位置或 IP 位置的地方进行报文的交换，因此大家也习惯了使用单个的交换机来管理多个物理机。但是在 ESX/ESXi 中，虽然也是一个 vSwitch 管理多个 VM，不过当多台 ESX/ESXi 组成集群之后，其上的 VM 常常属于同一个网段（使用同一个 Virtual Machine Group）时，还得要用多个 vSwitch 来管理，让整个 vCenter 环境充满了复杂性。



▲ 每一个 ESX/ESXi 上都有好几个虚拟交换机，配置起来相当麻烦

1. VI3 使用单主机的 vSwitch

由于 VI3 只有一种交换机，就是 vSwitch，因此所有的 ESX/ESXi 之间的数据交换都得通过配置所有 vSwitch。一个 vSwitch 又包括了 Service Console、不同的 VMkernel 以及让 VM 之间进行交换的 Virtual Machine Port Group。如果你只有几台 ESX/ESXi 服务器，那么在配置时只要注意一下之间的名称即可。但如果你要管理数十台以上的 ESX/ESXi，那么很可能就有上百个 vSwitch 要管理，这对系统管理人员来说是极大的负担。



▲ 每一个 VSS 上又有很多虚拟网卡，配置错误，VMotion 就不能用了

2. 传统的网络环境

传统的网络环境会使用单个交换机来管理多台主机，以上面的例子来说，多台 ESX/ESXi 的服务器一定是连到同一个物理交换机上，ESX/ESXi 和交换机之间用网络线连接，并且在交换机的底层进行 MAC 或 IP 的分组交换。我们不可能让一个 ESX/ESXi 主机使用一个交换机，再将这些多个交换机连接起来，那么为什么 VI3 要这么做呢？因此在 vSphere 时，VMware 提出了单个虚拟交换机的概念。



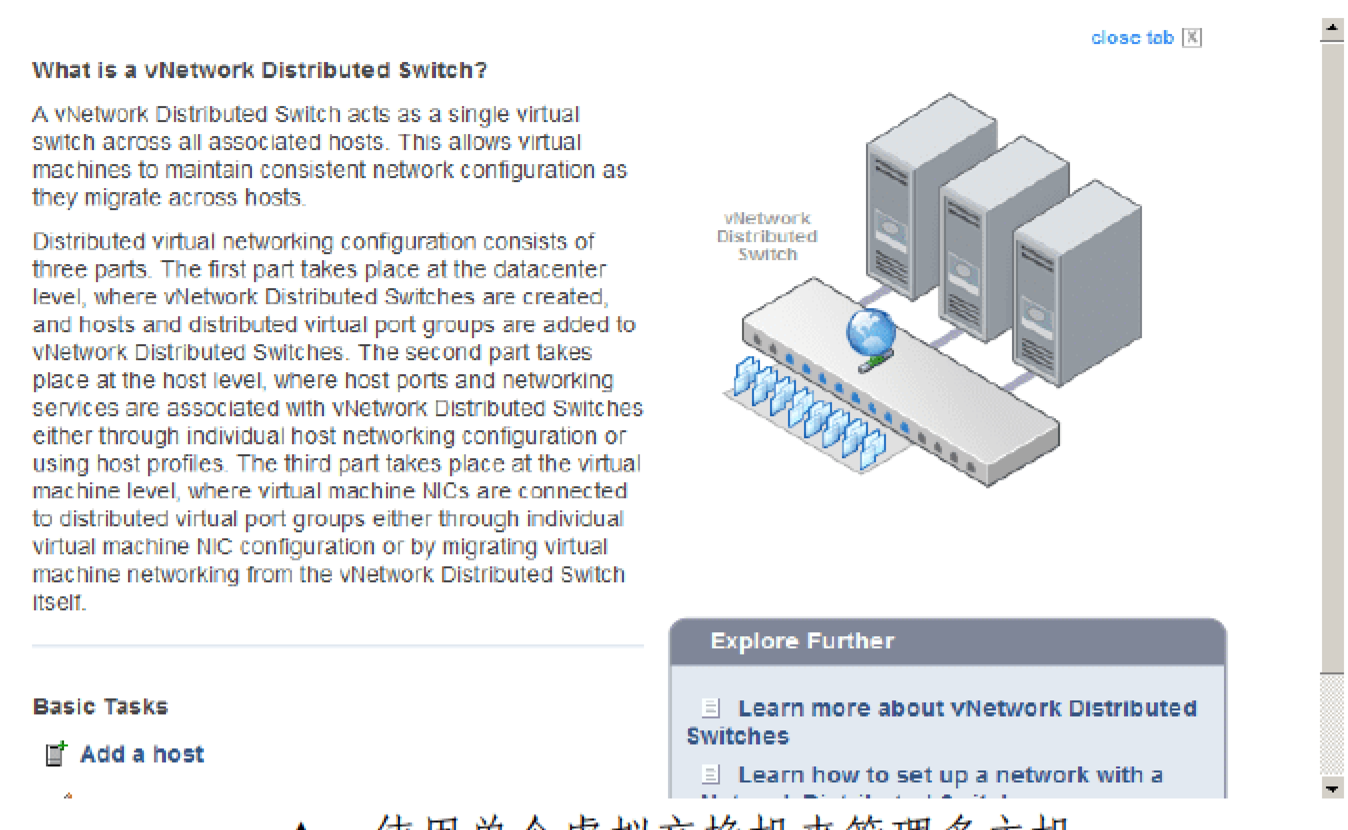
▲ 多主机本来就只有一个交换机

22.1.2 理解 vDS 的原理及功能

vDS 简单来说，就是以 vCenter 为中心，创建一个单个的虚拟交换机，这个虚拟交换机可以管理多台物理的 ESX/ESXi，而这个单个的虚拟交换机，就像物理交换机一样，可以处理物理主机之间的交通，并且也能充分发挥虚拟设备的特色，我们就来看看 vDS 的特色。

1. 单个交换机管理多 ESX/ESXi 主机

只要 vSphere 中的任何 ESX/ESXi 主机能联机到这台 vDS 上，就可以用这台 vDS 来管理。事实上，vDS 身为一个交换机，只要你的 ESX/ESXi 上有物理网卡，当然可以连上交换机，因此整个架构更像物理的交换机架构。



▲ 使用单个虚拟交换机来管理多主机

2. 使用 vCenter 管理

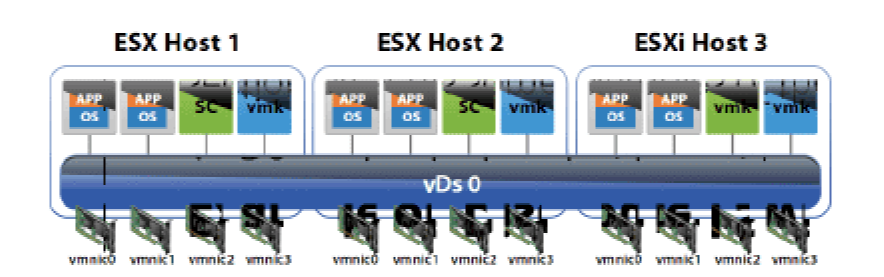
由于 vCenter 集合了多台 ESX/ESXi 主机，因此使用 vCenter 来操作 vDS 再方便不过了。事实上所有 vDS 的操作都必须通过 vCenter，vCenter 也提供了许多方便的管理界面来操作 vDS，本章稍后将会有完整介绍。



▲ 当然是使用 vCenter 管理

3. 多种交换模式使用

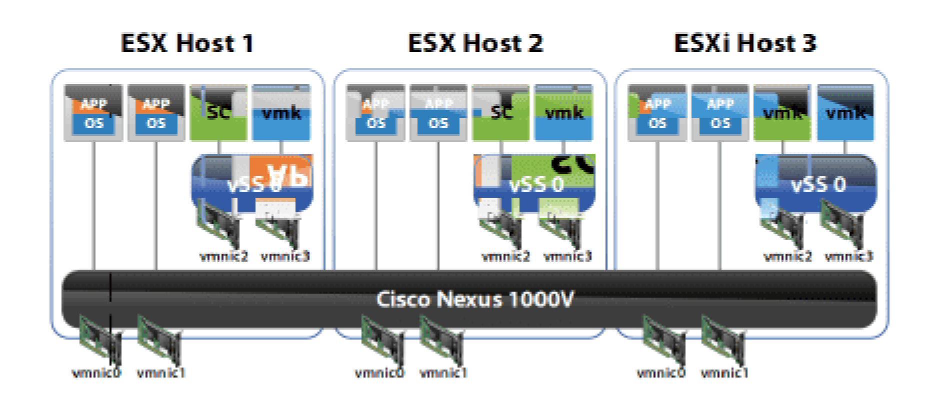
vDS 虽然简化了 vSphere 的网络，但 VSS 的使用也是不可或缺的。在一个较复杂的 vSphere 环境中，vDS 和 VSS 可以混合使用，你可以在一台 ESX/ESXi 中安装 VSS 来操作 Service Console 及 VMkernel，然后将 Virtual Machine 的端口放在 vDS，更可以在同一个 vSphere 环境中安装多个 Vds。



▲ 可以支持多种使用方式

4. 可以安装第三方的虚拟交换机

如果你觉得 vDS 不够用，别担心，vSphere 的 vDS 架构是容许你使用第三方虚拟交换机的，而全球最大的网络设备厂家思科（Cisco），在 vSphere 推出时，即发布了 Cisco Nexus 1000V 虚拟交换机，这是一个虚拟的 vDS，可以安装在 vSphere 环境中，我们在下一章会有完整的说明。

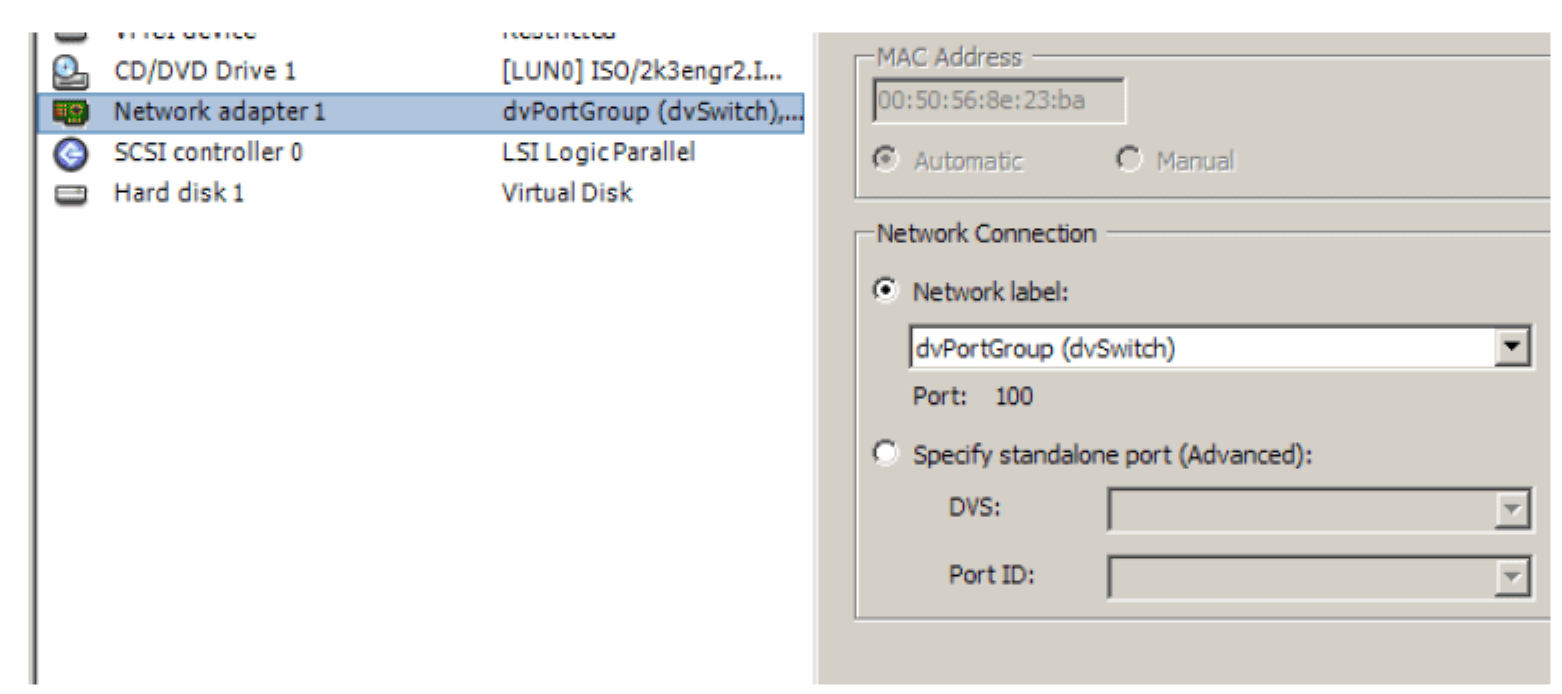


▲ 可使用第三方产品

5. 可将现成的 VSS“无痛”移植到 vDS 上

就算你现在的环境都是使用 VSS，我们也可以在完全不关闭 VM 的情况下，将 VSS 上所有

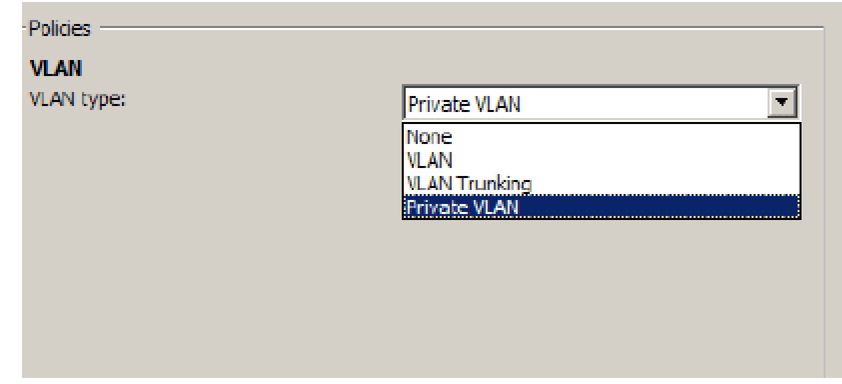
的功能移植到 vDS 上。这个称为 Network VMotion 的功能，正是 VMware 在推广 vDS 时最引以为傲的。在本章稍后也会有说明。



▲ 可以让 VM 无中断变更虚拟交换机

6. 各式各样的 VLAN 配置

vDS 支持 VLAN 已经不是什么新闻了。但 vDS 支持一个在 vSphere 中的新功能称为私人 VLAN（Private VLAN）。私人 VLAN 可以让 VM 集中在一些私有的网段，让整个网络不需要更多的子网段，方便 IT 人员的管理。



▲ 有多种的 VLAN 配置

7. 支持 Host Profile

我们在前面的章节提过 Host Profile 方便主机设置的移植。vDS 当然也支持 Host Profile。你可以配置好一台 ESX/ESXi 主机的 vDS 配置，再将这个配置导入其他的主机，让整个 vSphere 网络环境更加快捷地建置起来。

22.1.3 vDS 的基本部件

VMware 在推出了 vSphere 时，并发也推出了 vDS。vDS 的概念很简单，就是让多个 ESX/ESXi 主机能使用共同的虚拟交换机。这个虚拟交换机的功能和原来的 vSwitch 完全一样，也有 Service Console、不同的 VMkernel 以及让 VM 之间进行交换的 Virtual Machine Port Group。但除了这些功能之外，vDS 还得连接更底层的多个 ESX/ESXi，因此也需要进行到物理网卡层级的交换。我们就先来看看 vDS 的特色。

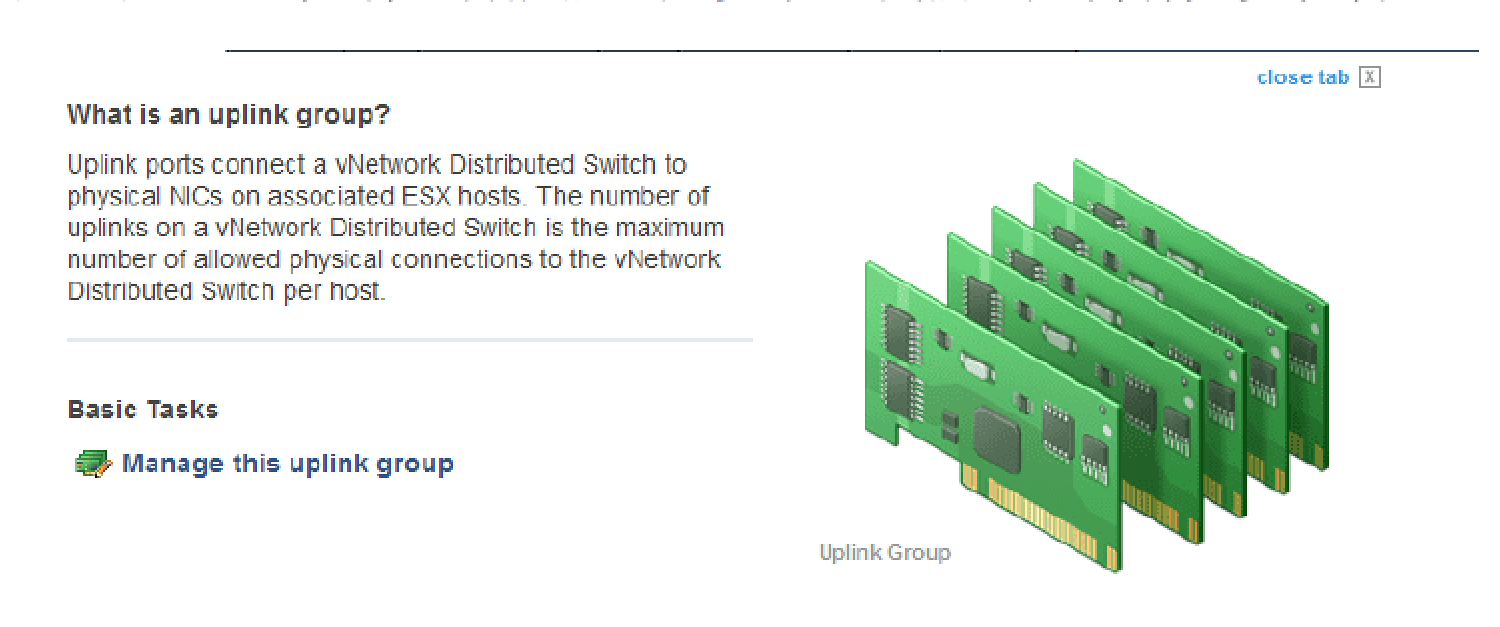
> **注意**
> **原来的 vSwitch 改名**
> 为了和原来的 vSwitch 之间有所区别，单独 ESX/ESXi 使用的 vSwitch 则称为 vNetwork Standard Swtich，简称为 VSS，以和 vDS 区别。我们在本章之中都以 VSS 意味着原来的 vSwitch。

vDS 一共有两个重要的部件，一个是和网卡连接的界面，另一个则是和 VSS 一样的 Port Group。

1. vDS 的连接界面：DVUplinks

vDS 必须和物理网卡连接，在 vDS 中称为 DV Uplink。当我们创建 vDS，系统第一件事就是

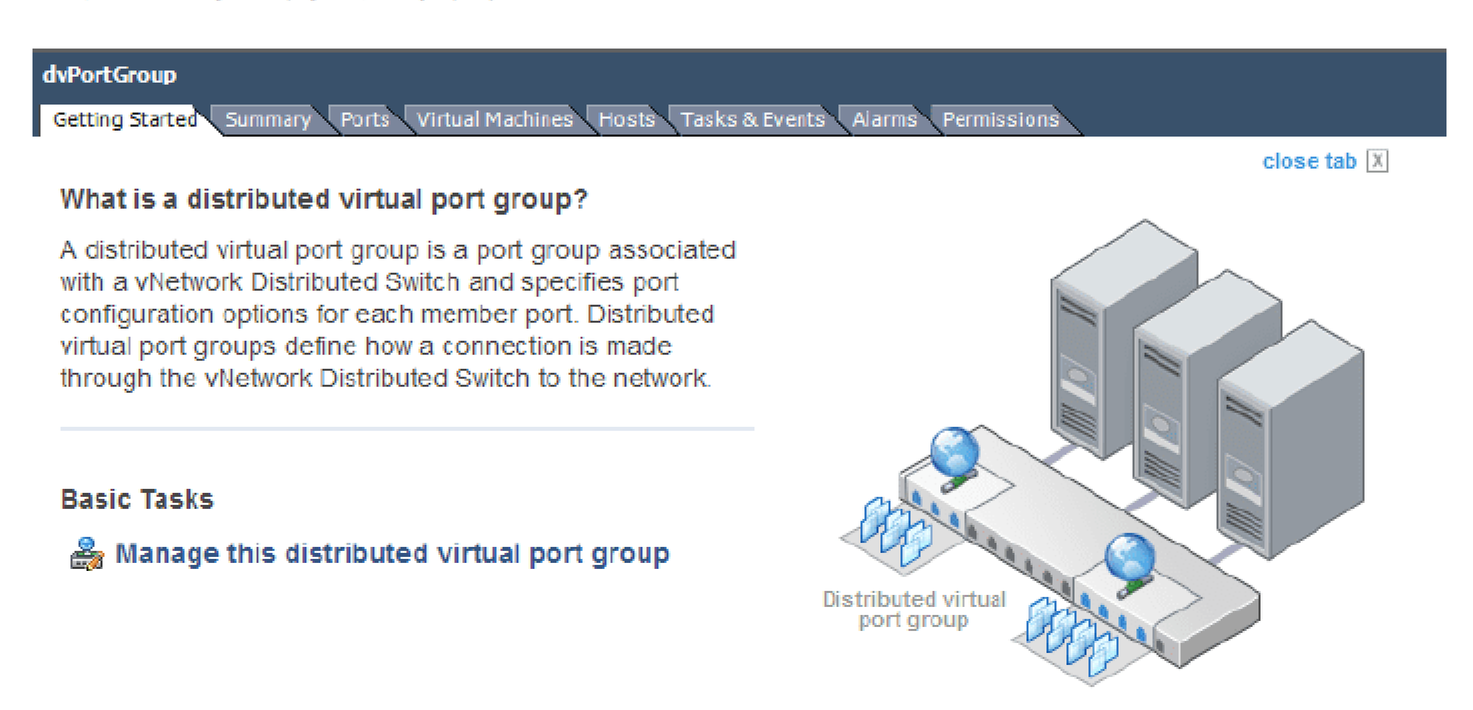
创建 DV Uplink。DV Uplink 事实上可以视为“交换机”上的“网络通信端口”。就像我们连接物理机时，也要将网络线插入交换机上的网络通信端口，计算机才能互相连接。当然 vDS 的 DV Uplink 自然比物理的网络通信端口功能强上太多了，我们之后会有更多说明。



▲ 这是虚拟的通信端口

2. vDS 的通信端口群组 dvPortGroup

vDS 和 VSS 有很多地方是完全一样的，只是 vDS 是针对多主机，而 VSS 是针对单个主机。Port Group 就是一个最好的例子。在 vDS 中，我们可以创建 Service Console、VMkerne 和 VM port groups。我们在 vDS 中如果要实现这些功能，就必须在 dvPortGroup 中落实，当然落实的手段和 VSS 大同小异，只是当主机变多之后，主机之间的通信是使用网卡而非虚拟交换机了，但这些改变只会让 vSphere 的网络设计更加简单。



▲ 这是具体连接的地方

22.2 具体操作 vDS

很多对 VI3 很熟的用户，在 vSphere 中反而觉得 vDS 太复杂。事实上完全相反。由于 VSS 的单主机配置，让多 ESX/ESXi 的环境必须每次都让每一个功能重新来过。因此如果你已经对 VSS 相当熟练，不妨放下这个旧产品，全面迎接单个虚拟交换机的时代，我们在这一小节中，就来看看 vDS 的使用。

22.2.1 创建 vDS 及其部件

在创建 vDS 之前，我们首先要有一个观念，就是 vDS 是一个物理 ESX/ESXi 之间的交换机，

因此在真正的网络结构中，必须具有下面的前提。

▶ 创建 vDS 的前提

（1）所有加入这个 vDS 的 ESX/ESXi 主机必须都在同一真正的交换机上，使用物理交换机的 VLAN 是允许的。

（2）要加入 vDS 的 ESX/ESXi 主机必须以物理网卡的身份加入，如果有多余的网卡，可以直接加入，如果网卡已经被 VSS 占用，必须用移植的方式。

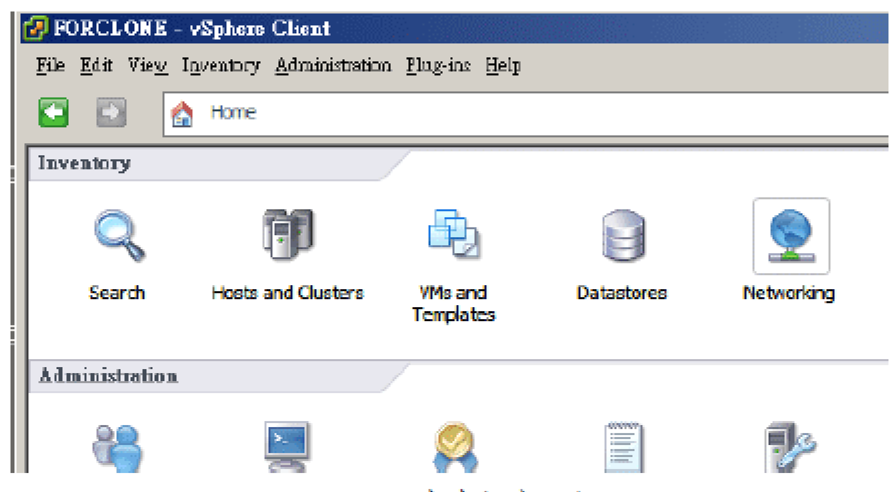
（3）所有要加入 vDS 的 ESX/ESXi 主机必须能被 vCenter 管理。

1. 创建新的 vDS

创建新的 vDS 十分简单，只要照着下面的步骤进行即可。但要注意的是，在创建新的 vDS 时，vSphere 会并发也创建 DVUplink 以及 dvPortGroup。如果你在创建 vDS 时选择不创建 dvPortGroup，之后将 VM 放入时可能会无法从 DHCP 取得 IP，因此在这里一定要并发创建。

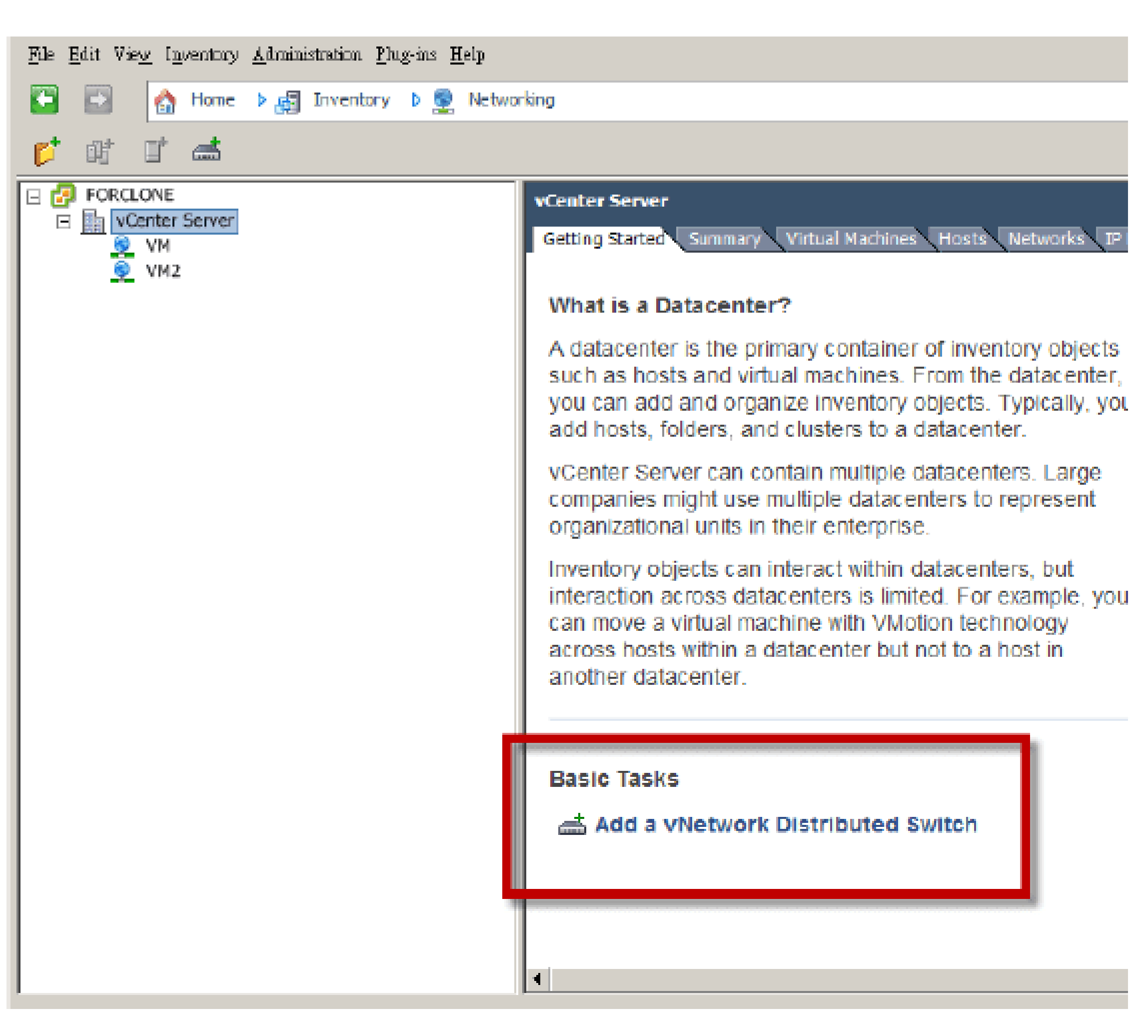
▶ 创建新的 vDS

1. 首先进入 vCenter 中 并且选择 Home/Networking 链接。



▲ 选择这里

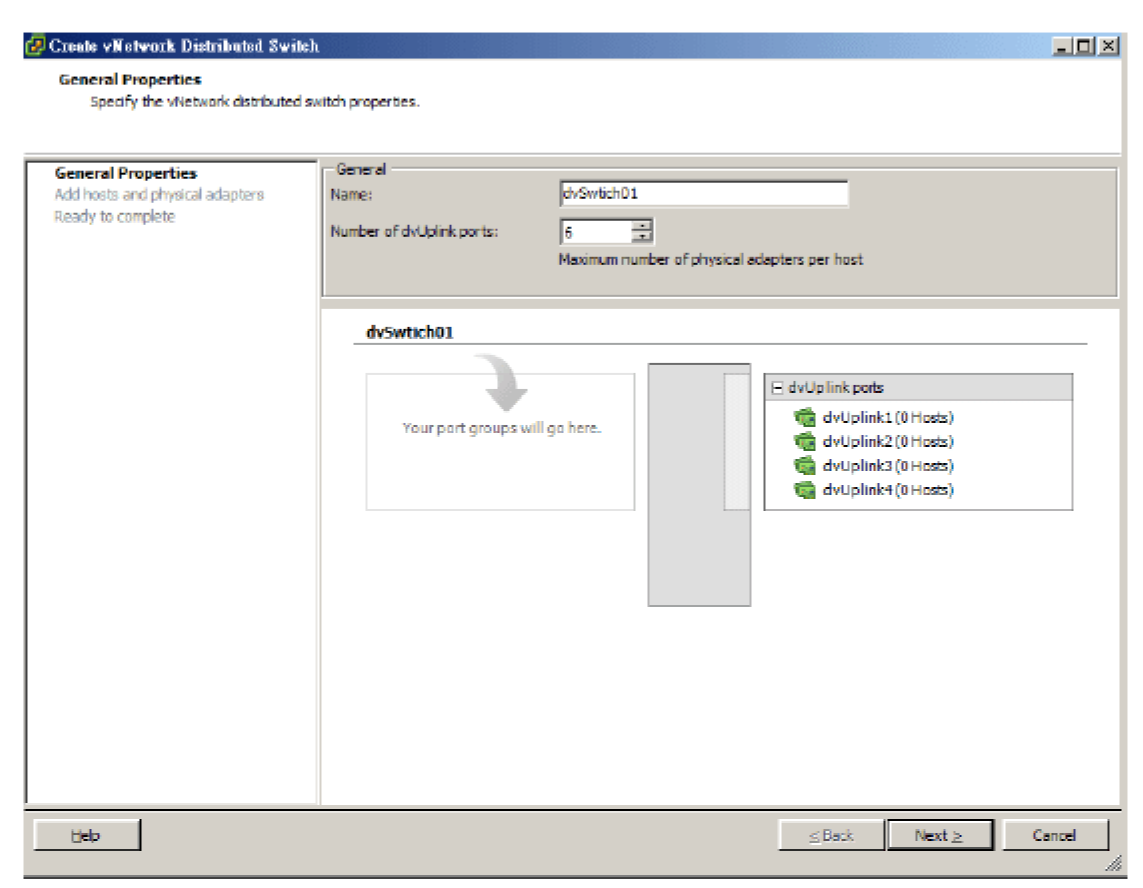
2. 此时会弹出当前的 VM 交换机，当前只有两个 VM。选择 Add a vNetwork Distributed Swtich 链接。



▲ 新建 vDS

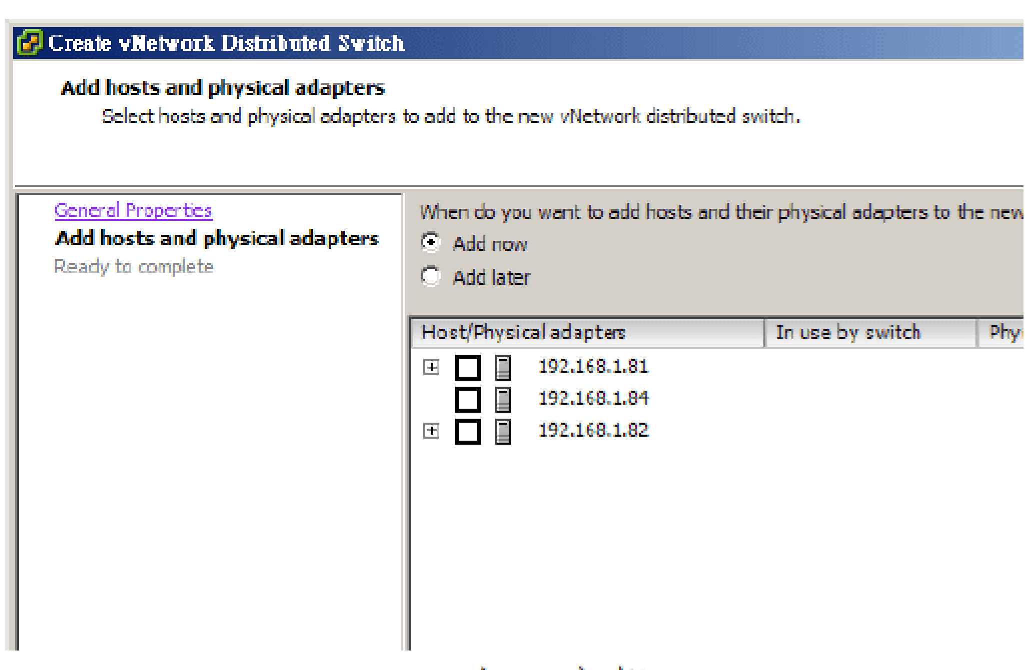
3. 接下来会弹出 vDS 创建画面。首先要替这个 vDS 取个名字。接下来是最大的 dvUplink

ports 的数量。这个数量就是这个 vDS 能连接物理网卡的最大数量，你可以将这个数字看成一个物理交换机上的端口数量。单击 Next 按钮继续。



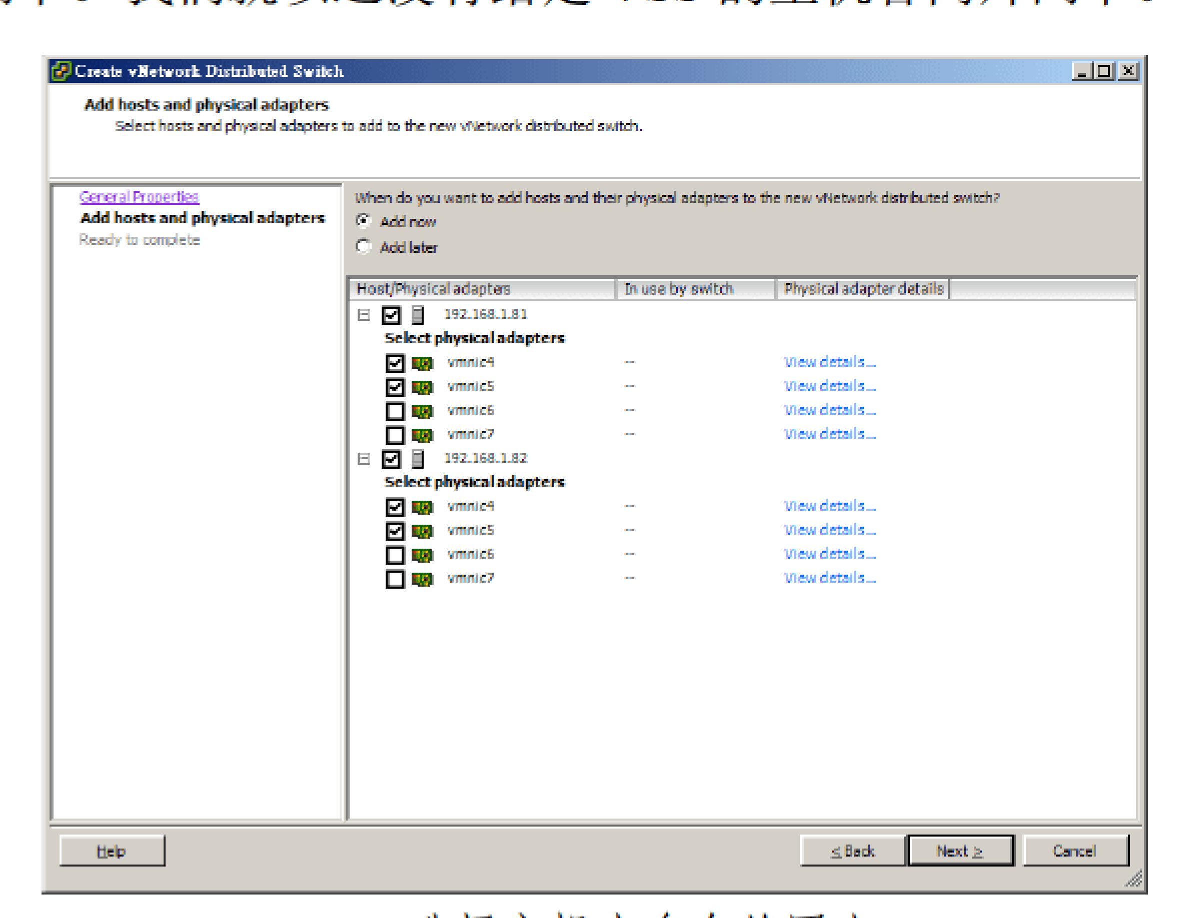
▲ 创建新 vDS

4. 接下来是将 ESX/ESXi 主机加入 vDS 中。系统会给出这个 vCenter 管理的 ESX/ESXi 主机。



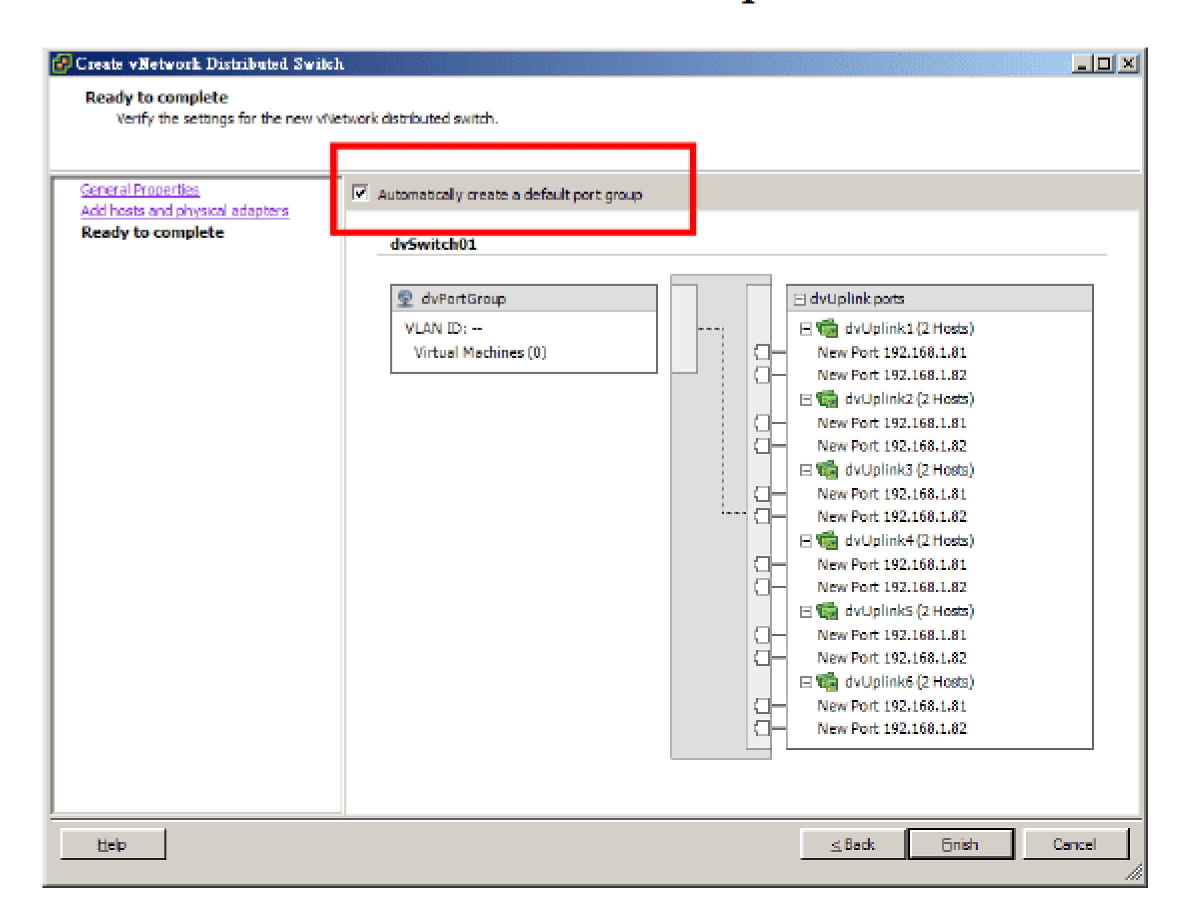
▲ 加入主机

5. 我们单击打开主机的“+”卷展符，可以看到 ESX/ESXi 的网卡。选择要“插上”这个交换机的网卡。我们就以还没有给定 VSS 的主机各两片网卡。单击 Next 按钮继续。



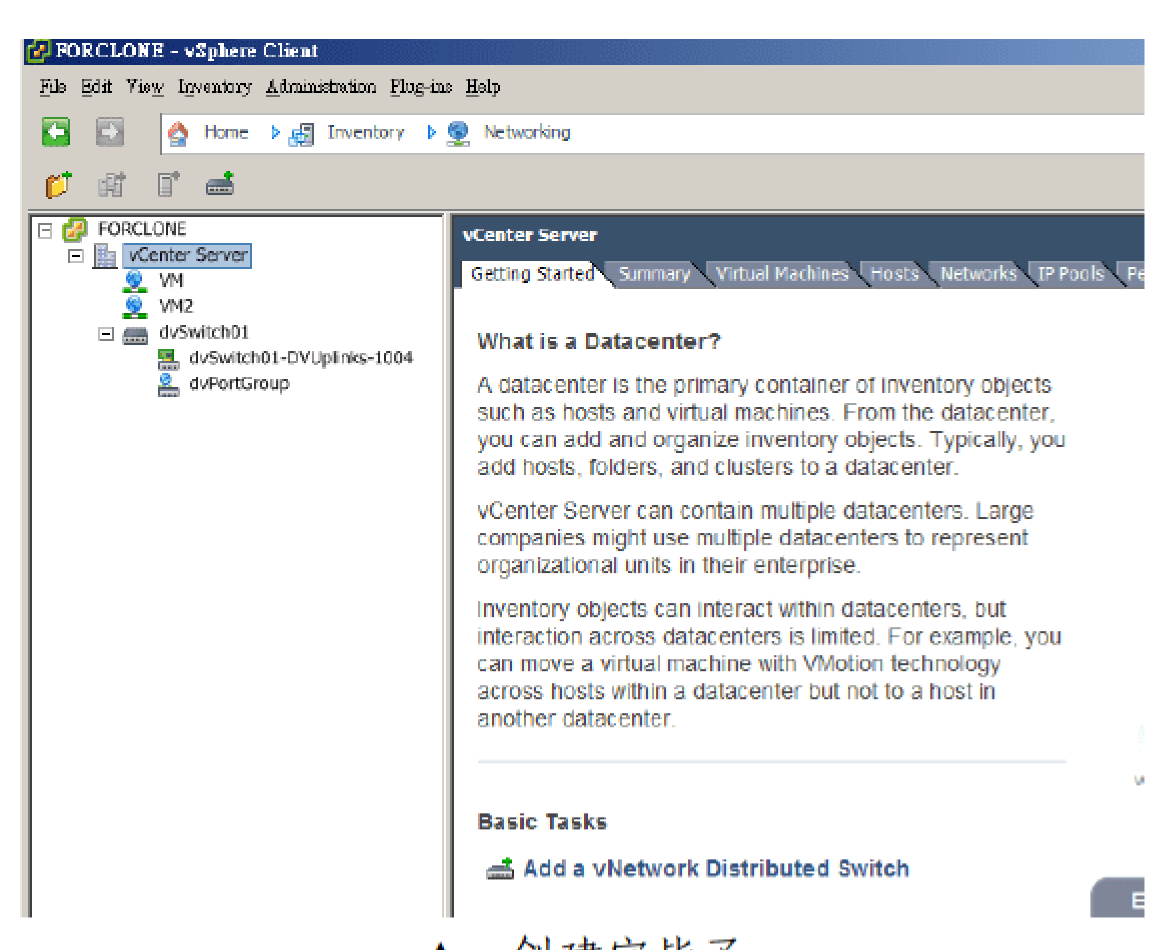
▲ 选择主机上多余的网卡

6. 接下来我们会要求系统顺便创建 dvPortGroup。单击 Finish 按钮继续。



▲ 在创建时一定顺便创建 dvPortGroup，之后创建的无法使用

7. 此时我们会看到新建的 vDS 其下的 DVUplink Ports 和 dvPortGroup。



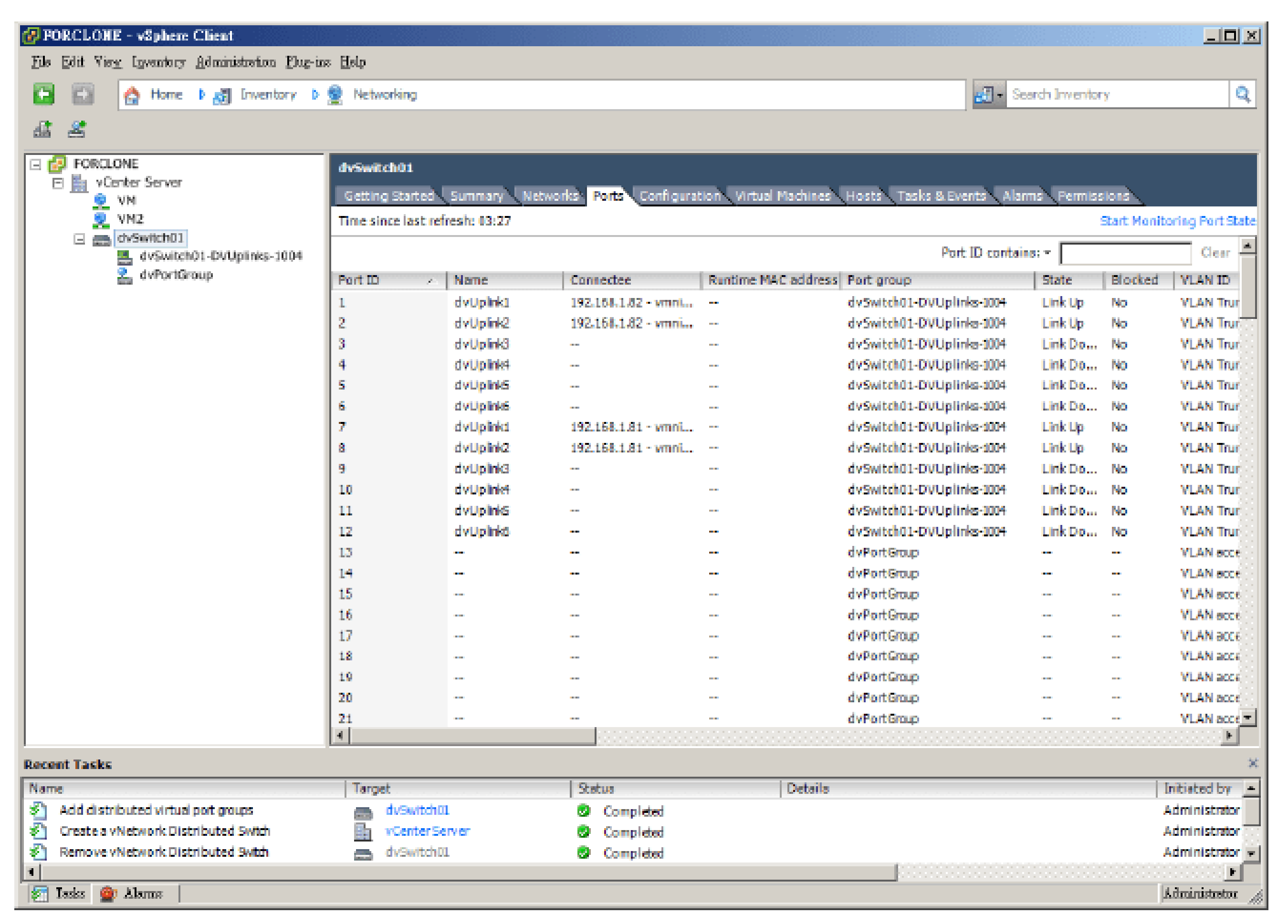
▲ 创建完毕了

2. 配置你的 vDS

当我们创建好 vDS 之后，其实大部分的事情都已经做完了。我们可以看看这个 vDS 不同的 Port 的配置。

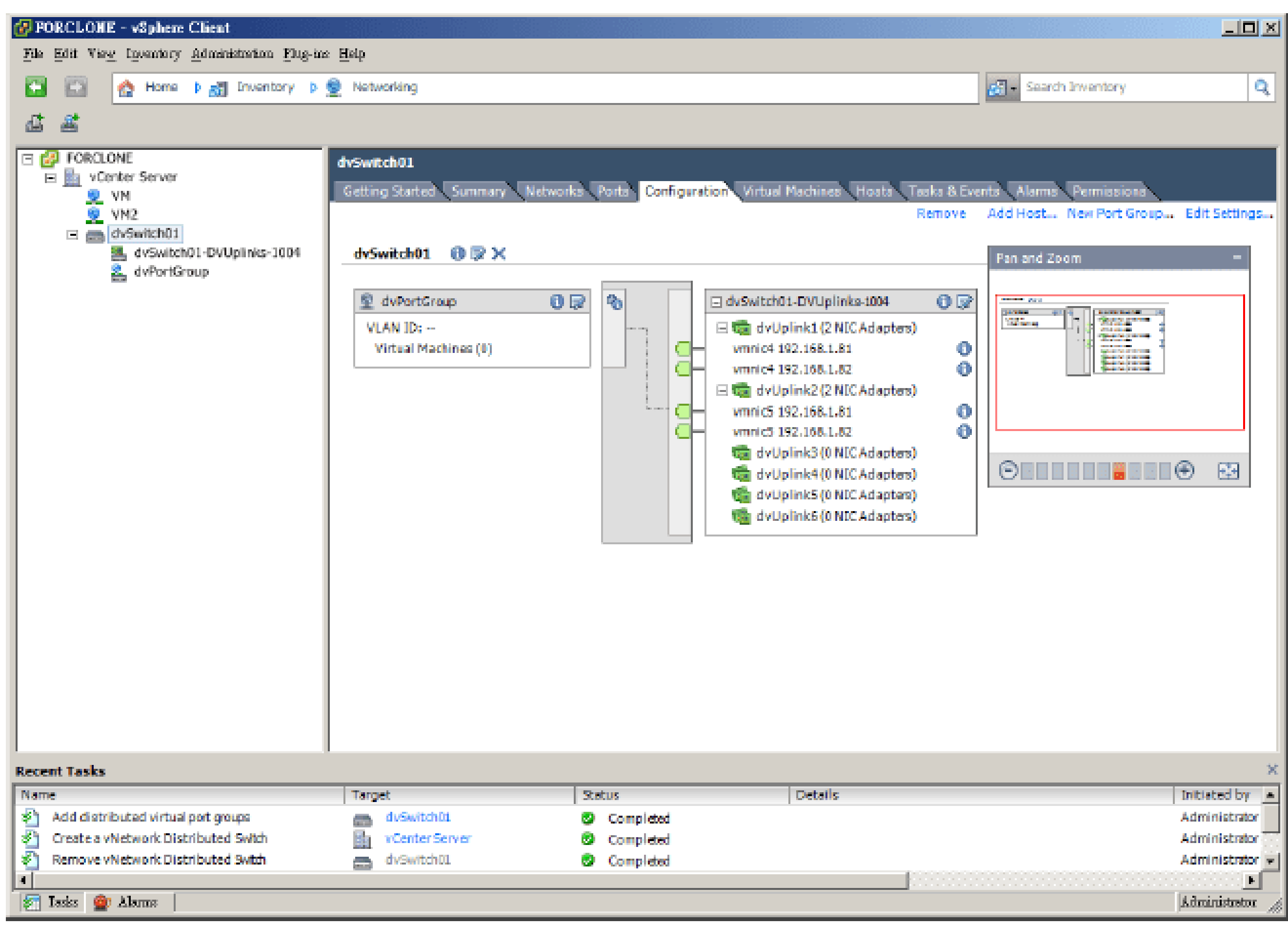
▶ 理解 vDS 不同 Port 的配置及使用

1. 将游标放在 dvSwitch01 上，并且移到 Ports，可以看到当前这个 vDS 的 DVUplink Ports 和主机网卡连接的情况。



▲ 和主机连接的情况

2. 在 Configuration 的选项卡中，可以看到 dvPortGroup 的连接情况。当前还没有任何 VM 连上这个 vDS，也没有任何 VSS 的 VMkernel 功能对应过来。



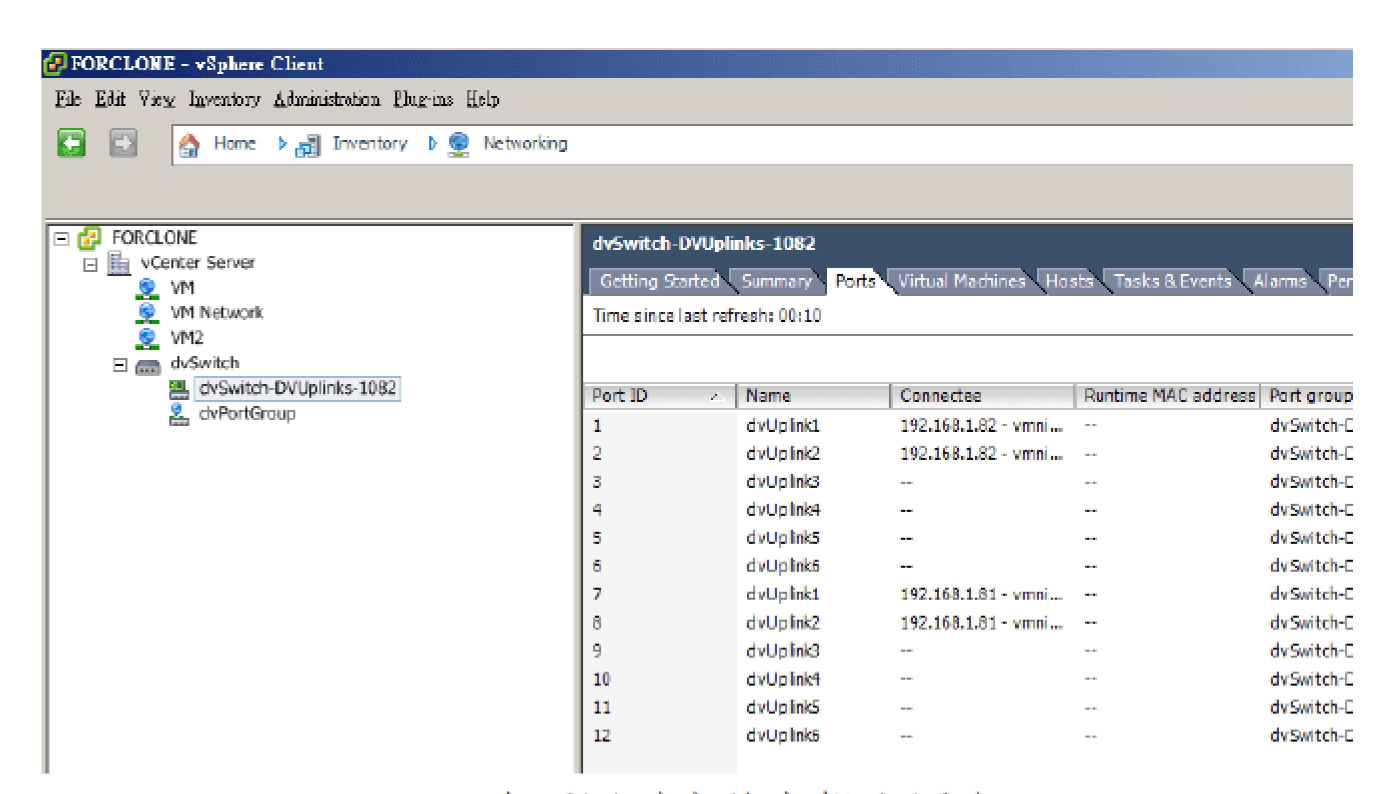
▲ 当前没有 VM 连接

3. 在 Host 选项卡中，可以看到两台加入的 ESX/ESXi 主机。



▲ 新加入的两台 ESX

4. 将游标移到创建的 DVUplink 上，也可以看到不同的对应。一般来说，vDS 在给定了 VM 之后的作用较大。



▲ 有不同对应的主机和网卡

22.2.2 VM 中的操作及配置

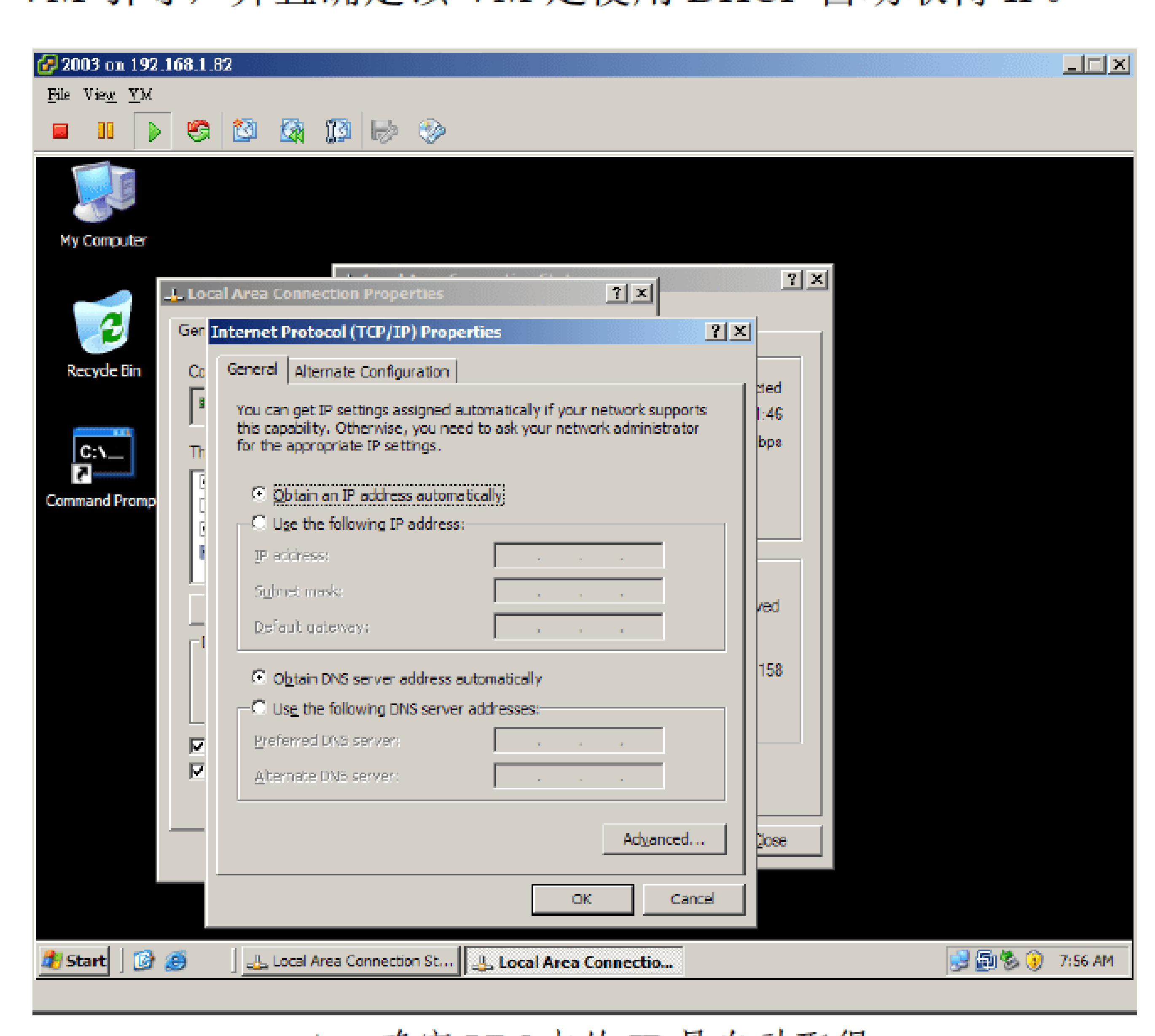
要将原来使用 VSS 的 VM 转换到 vDS 十分简单，而在转换的过程中，整个网络是不会停止的。我们就来看看如何让 VM 从使用 VSS 中的 Port Group 转而使用 vDS 中的 dvPort Group。

1. 使用 Network VMotion

Network VMotion 是 vSphere 中一项特殊的功能，但配置起来却相当简单，我们就来看看如何让一台 VM 在不关机的情况之下，从 VSS 中的 VM Port Group 移到 vDS 的 dvPortGroup，但仍然保持 IP 不变，甚至网络都不会中断！

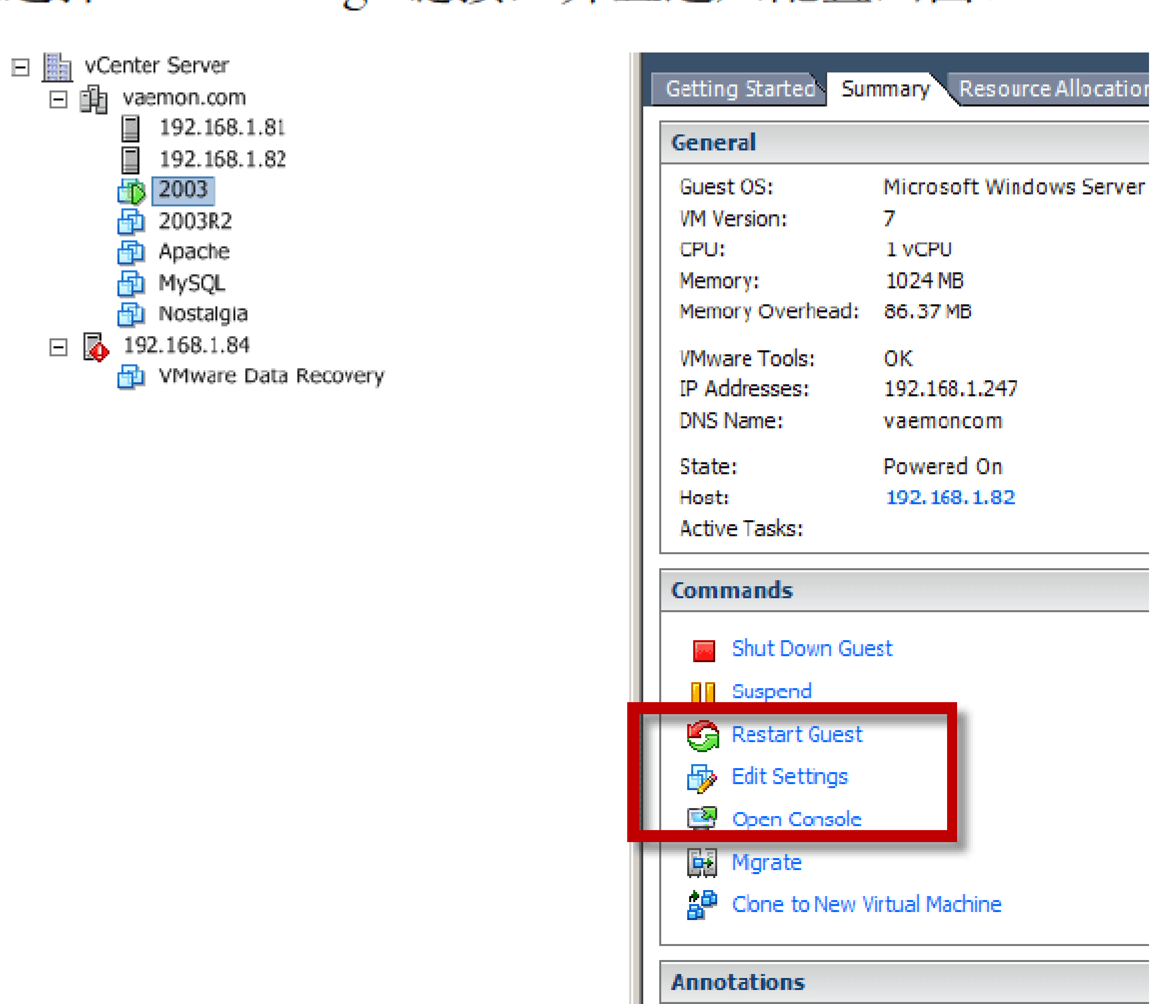
▶ 使用 Network VMotion

1. 首先将 VM 引导，并且确定该 VM 是使用 DHCP 自动取得 IP。



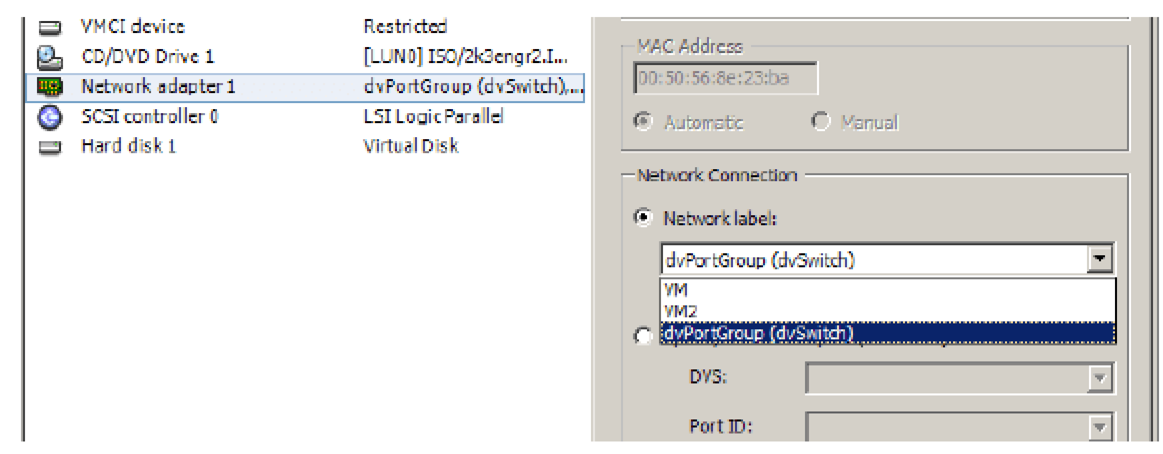
▲ 确定 VM 中的 IP 是自动取得

2. 确定 VM 存在的 ESX/ESXi，是已经加入 vDS 的。
3. 在该 VM 上选择 Edit Settings 链接，并且进入配置画面。



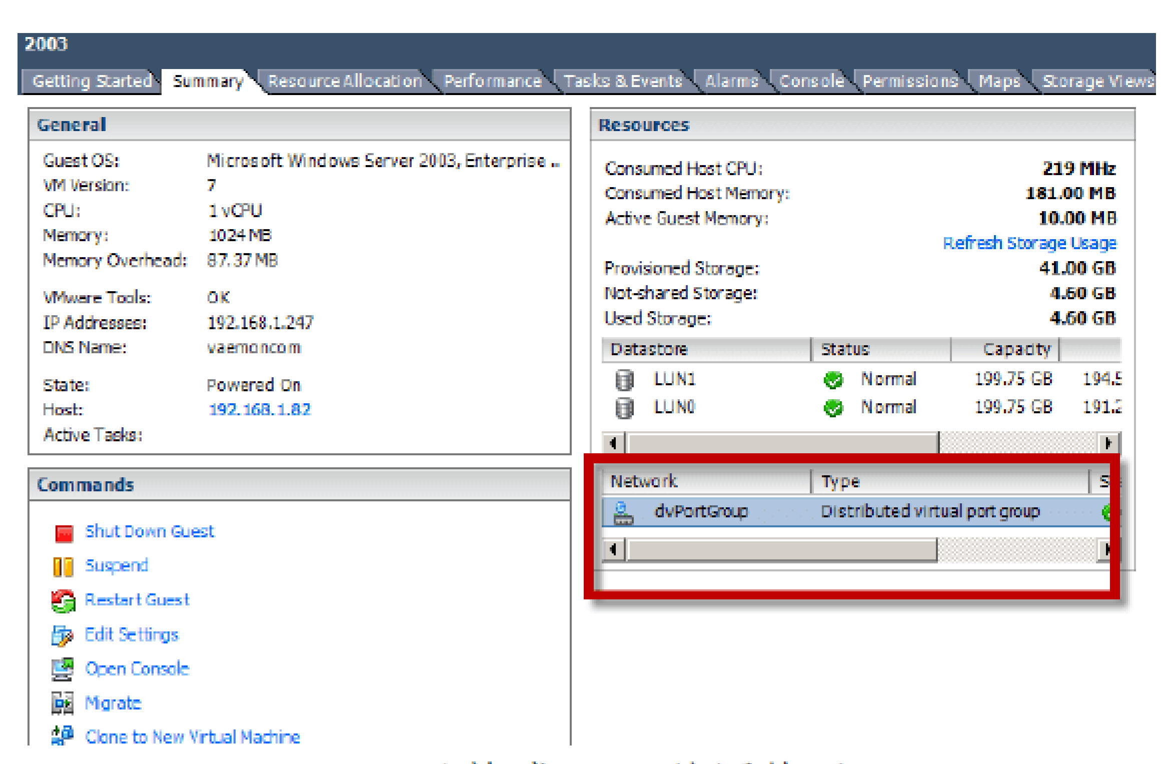
▲ 变更 VM 的配置

4. 此时我们找到该 VM 的网卡，并且将原来的 VM 换成 vDS 的 dvPortGroup。之后单击 OK 按钮。



▲ 更换 VM 的网络

5. 此时我们回到 VM 的 Summary 画面，会发现原来的 Standard switch network 已经变成 Distributed virtual port group 了。



▲ 已换成 vDS 的网络了

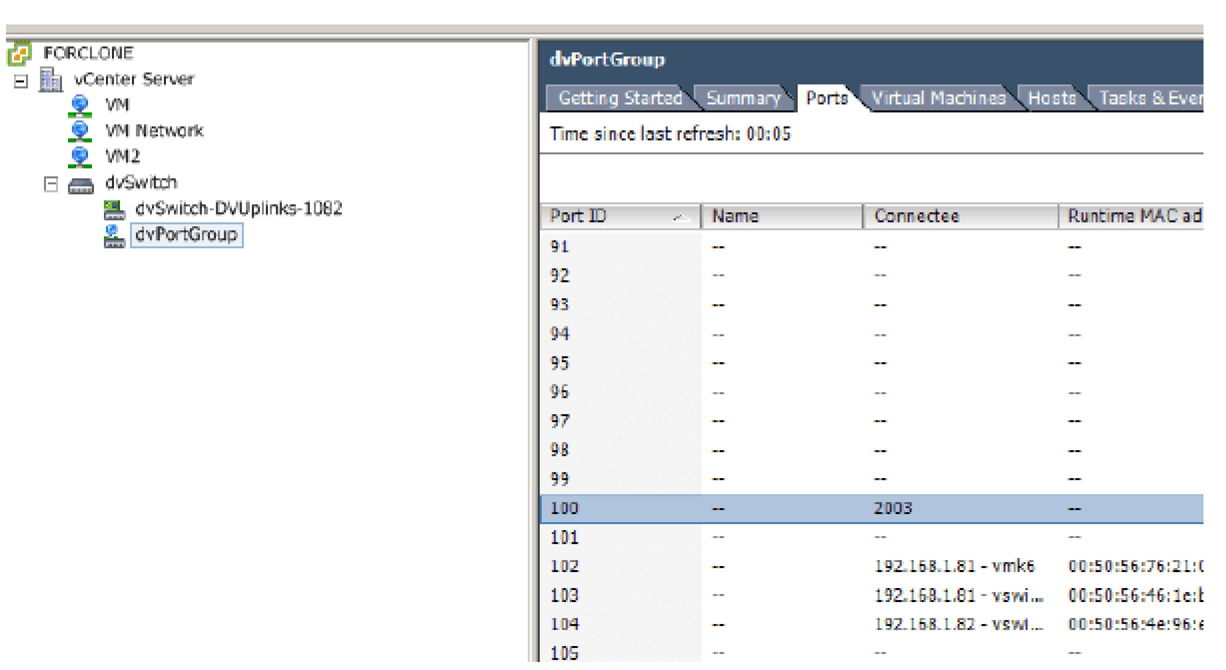
6. 在整个变换的过程中，我们会发现 VM 的 Ping 从来没有断过，一个报文都没有掉。

```
Reply from 192.168.1.247: bytes=32 time=1ms TTL=128
Reply from 192.168.1.247: bytes=32 time=1ms TTL=128
Reply from 192.168.1.247: bytes=32 time<1ms TTL=128
Reply from 192.168.1.247: bytes=32 time=1ms TTL=128
Reply from 192.168.1.247: bytes=32 time<1ms TTL=128
Reply from 192.168.1.247: bytes=32 time=1ms TTL=128
Reply from 192.168.1.247: bytes=32 time=1ms TTL=128
Reply from 192.168.1.247: bytes=32 time=1ms TTL=128
Reply from 192.168.1.247: bytes=32 time=1ms TTL=128
Reply from 192.168.1.247: bytes=32 time<1ms TTL=128
Reply from 192.168.1.247: bytes=32 time<1ms TTL=128
Reply from 192.168.1.247: bytes=32 time<1ms TTL=128
Reply from 192.168.1.247: bytes=32 time<1ms TTL=128
Reply from 192.168.1.247: bytes=32 time<1ms TTL=128
Reply from 192.168.1.247: bytes=32 time=1ms TTL=128
Reply from 192.168.1.247: bytes=32 time=1ms TTL=128
```

▲ 整个报文都没有掉

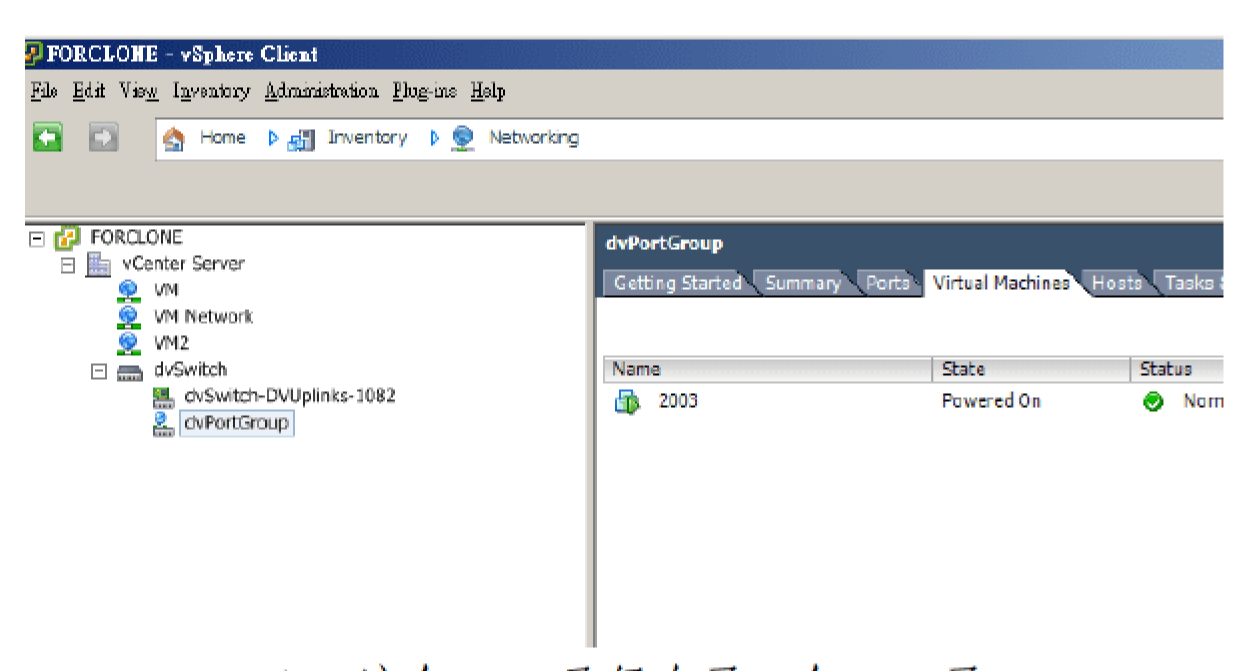
这个动作其实就像将一台主机的网络线从某个交换机拔下来，再插上另一个交换机，但之间的断线时间几乎是 0 秒。

7. 此时再进入 Home/Networking，进入该 vDS 的 dvPortGroup，可以看到在 Ports 的选项卡中已经有这个 VM 占用一个了。



▲ 已经占用一个 Port

8. 在 Virtual Machine 的选项卡也看到 2003 这台 VM。



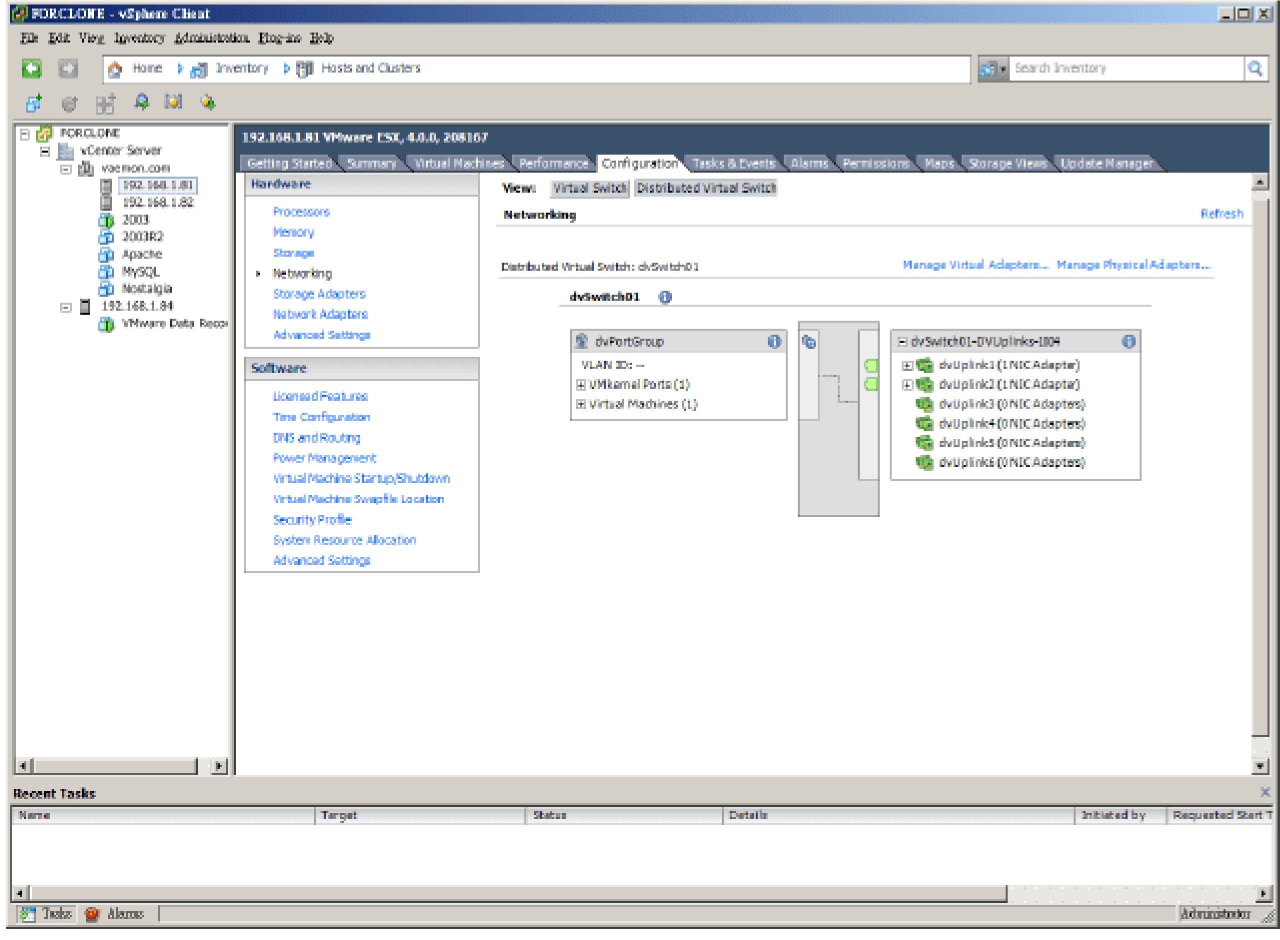
▲ 这个 vDS 已经有了一个 VM 了

2. 管理 ESX/ESXi 的物理网卡

如果你的 ESX/ESXi 在加入 vDS 之后，又插上了新的网卡，才可以将这片新的网卡加入 vDS 中。下面就是步骤。

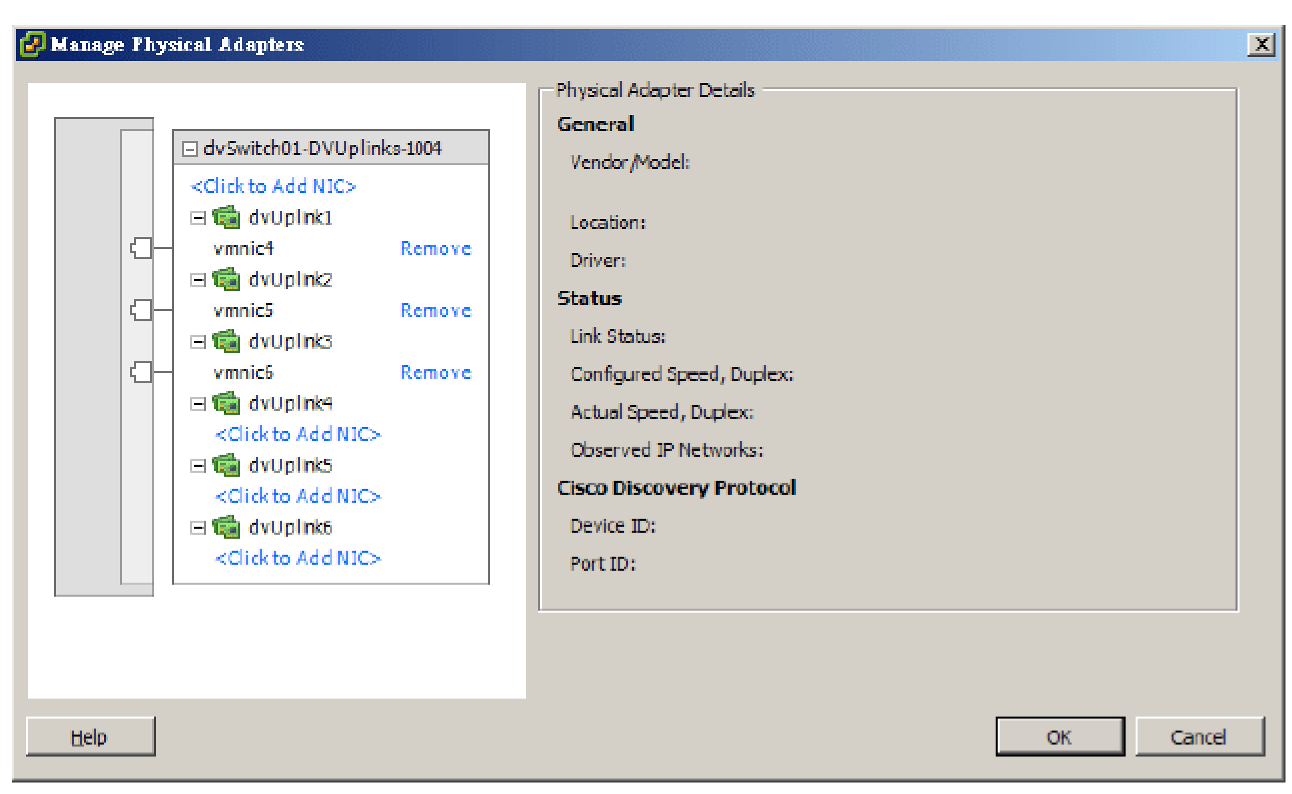
▶ 加入新的网卡

1. 进入 vCenter 中，并且选择要加入 vDS 的 ESX/ESXi 主机。选择 Configuration/Networking/Distributed Virtual Switch。并且选择“Manage Physical Adapters”。



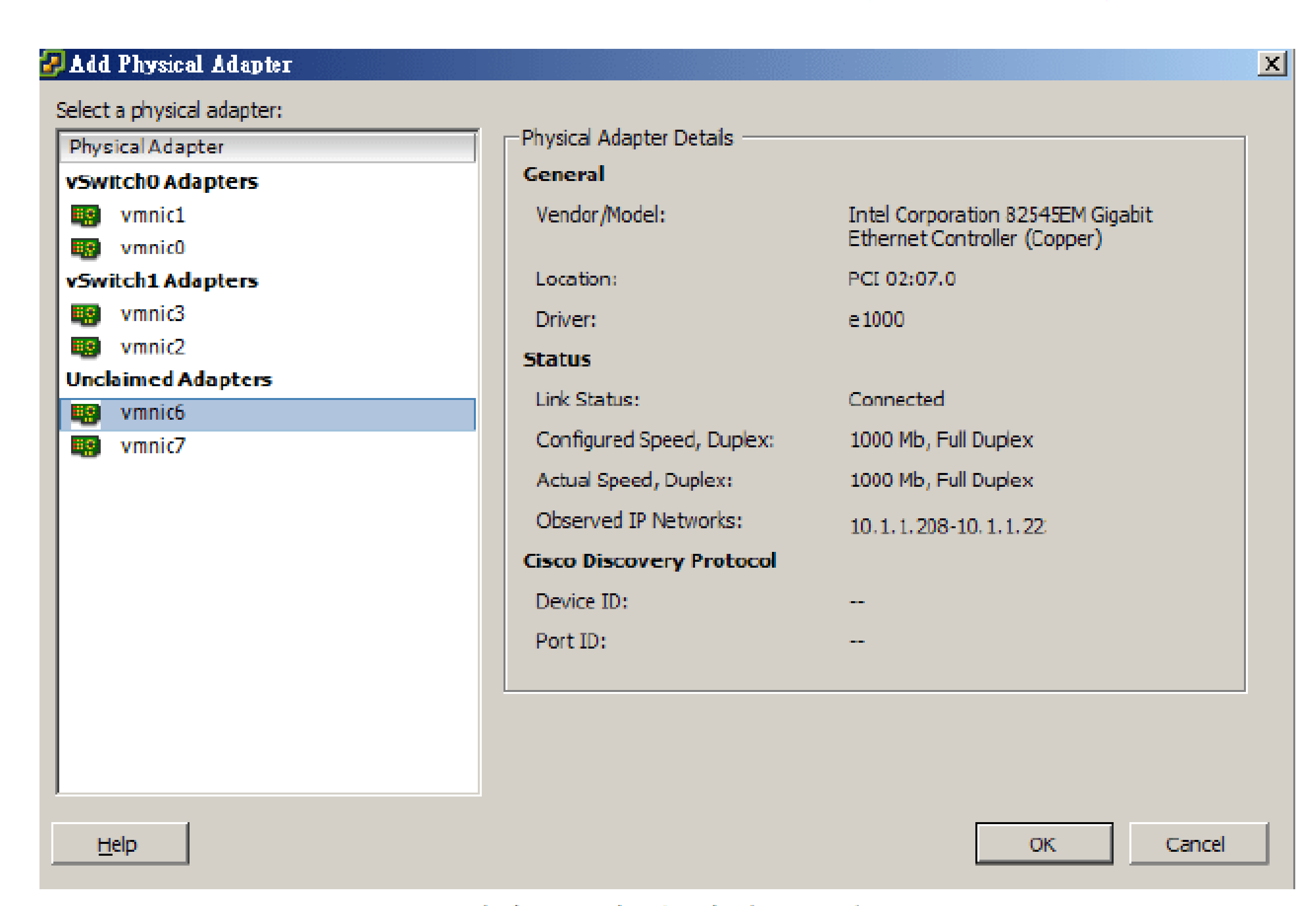
▲ 选择这里

2. 此时会给出已经加入 vDS 和还可以加入的 vDS 通信端口。如图中的 dvUplink3/4/5/6，可以单击 Click to Add NIC 加入。

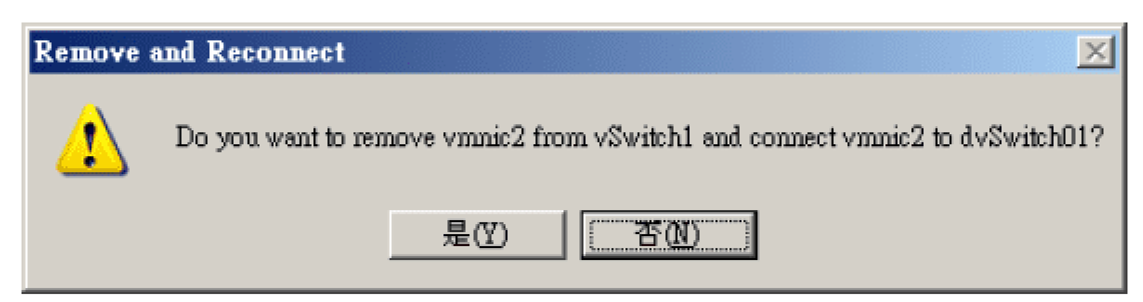


▲ 会给出可加入的 NIC

3. 此时可以选择没有使用的网卡加入，或是已经使用的网卡加入。如果是使用已经分配到 VSS 的网卡，就会弹出报警窗口。我们选择一个没有使用的网卡。

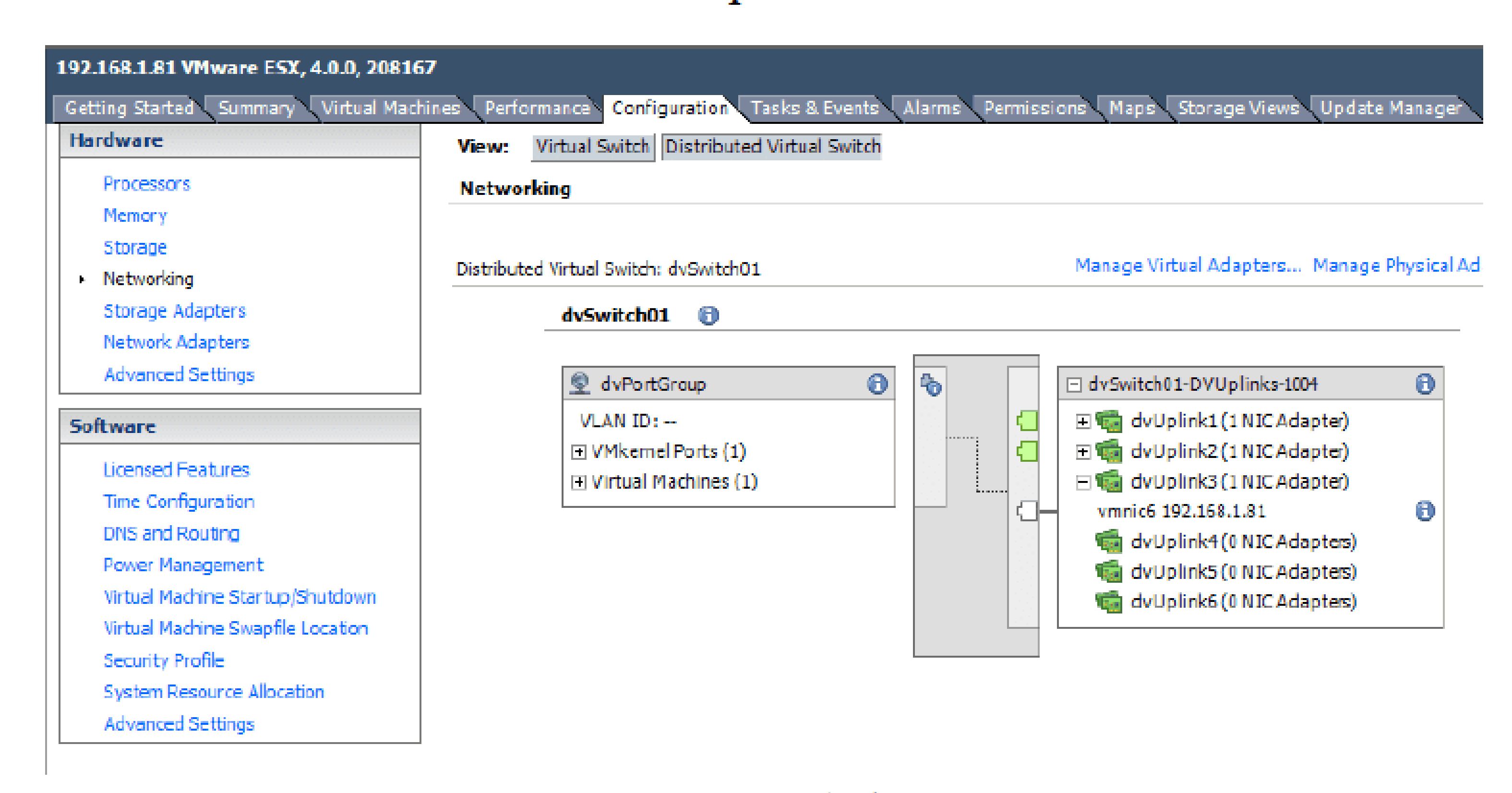


▲ 选择一个没有加入的 NIC



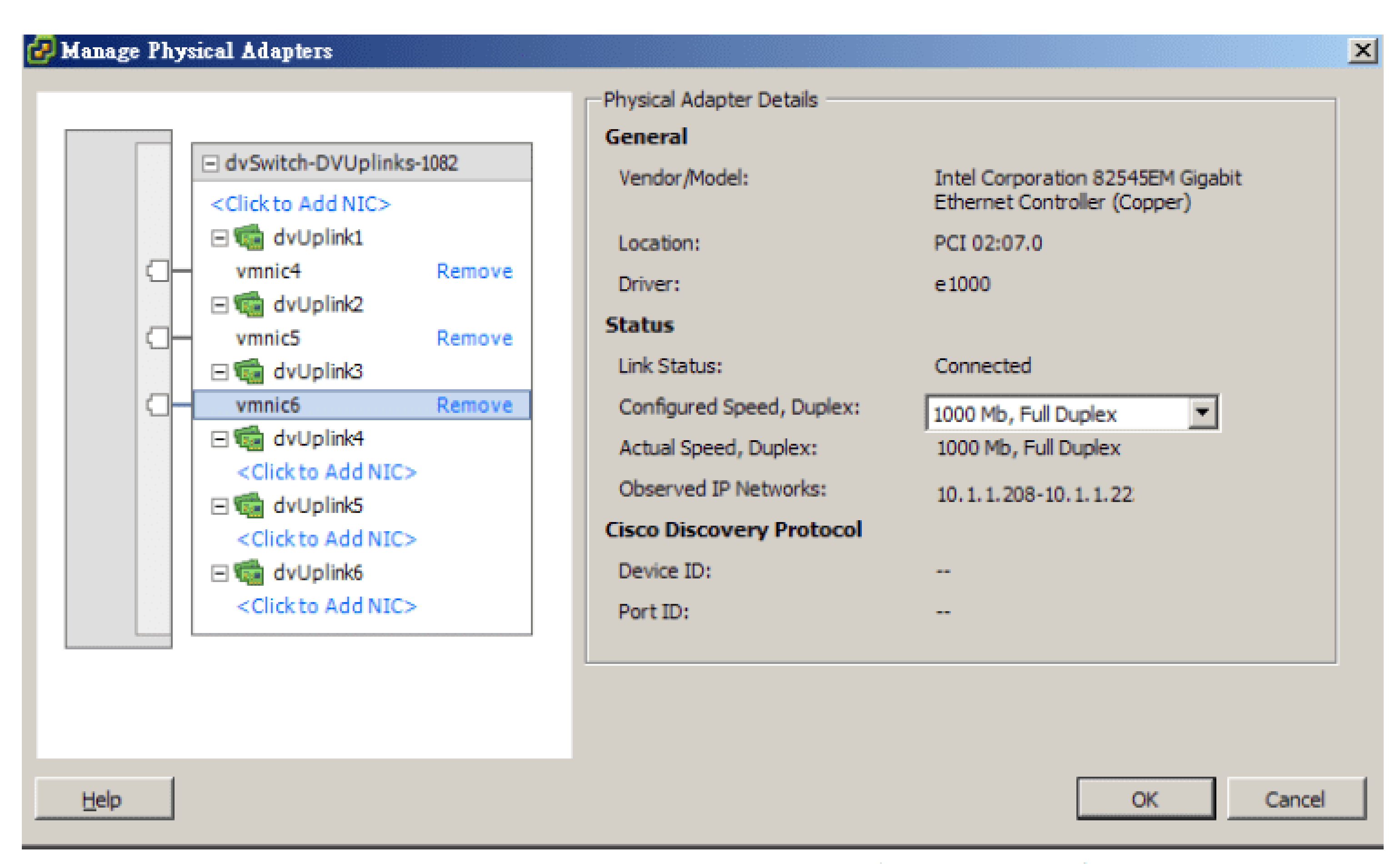
▲ 如果已经有 VSS 对应的 NIC，会有报警窗口

4. 接下来就看到 vmnic6 已经加入 dvUplink3 了。单击 OK 按钮，系统就会进行补丁。



▲ vmnic6 已经加入了

5. vmnic6 已经加入到 dvUplink3。



▲ 将 ESX 的物理网卡“插上”虚拟的交换机了

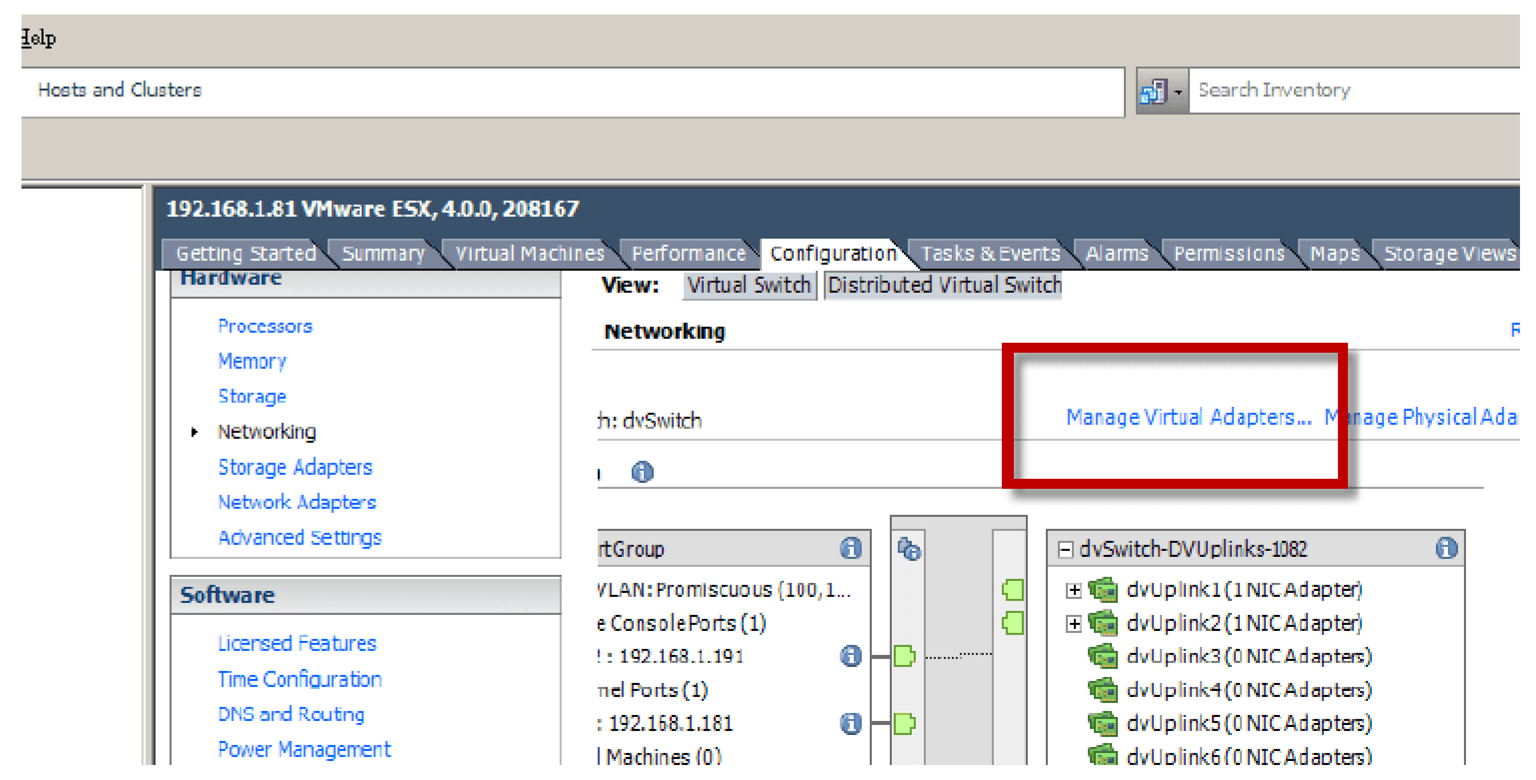
3. 用虚拟网卡管理 ESX/ESXi 的网络功能

既然 vDS 也是交换机，VSS 也是交换机，那么既然 vDS 能作为 VM 的 Port Group，能不能作为 Service Console、VMkernel 等功能呢？答案当然是肯定的。不过在 vSphere 中，交换机必须靠网卡相连接，因此要实现 Service Console、VMkernel 等功能，也必须通过网卡，但是这些功能则是使用虚拟网卡来落实。

物理网卡连接的是 dvUplink，而虚拟网卡则是连接到 dvPortGroup，因此我们在新建 ESX/ESXi 上的虚拟网卡时，目的就是要新建 PortGroup，用来作为 Service Console 及 VMkernel。下面我们就来看看具体做法。

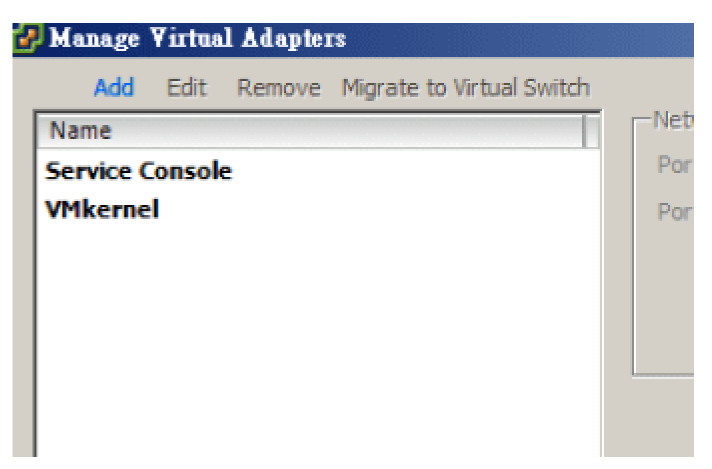
▶ 使用虚拟网卡新建网络功能

1. 进入vCenter中，并且选择要加入vDS的ESX/ESXi主机。选择Configuration/Networking/Distributed Virtual Switch。并且选择 Manage Virtual Adapters。



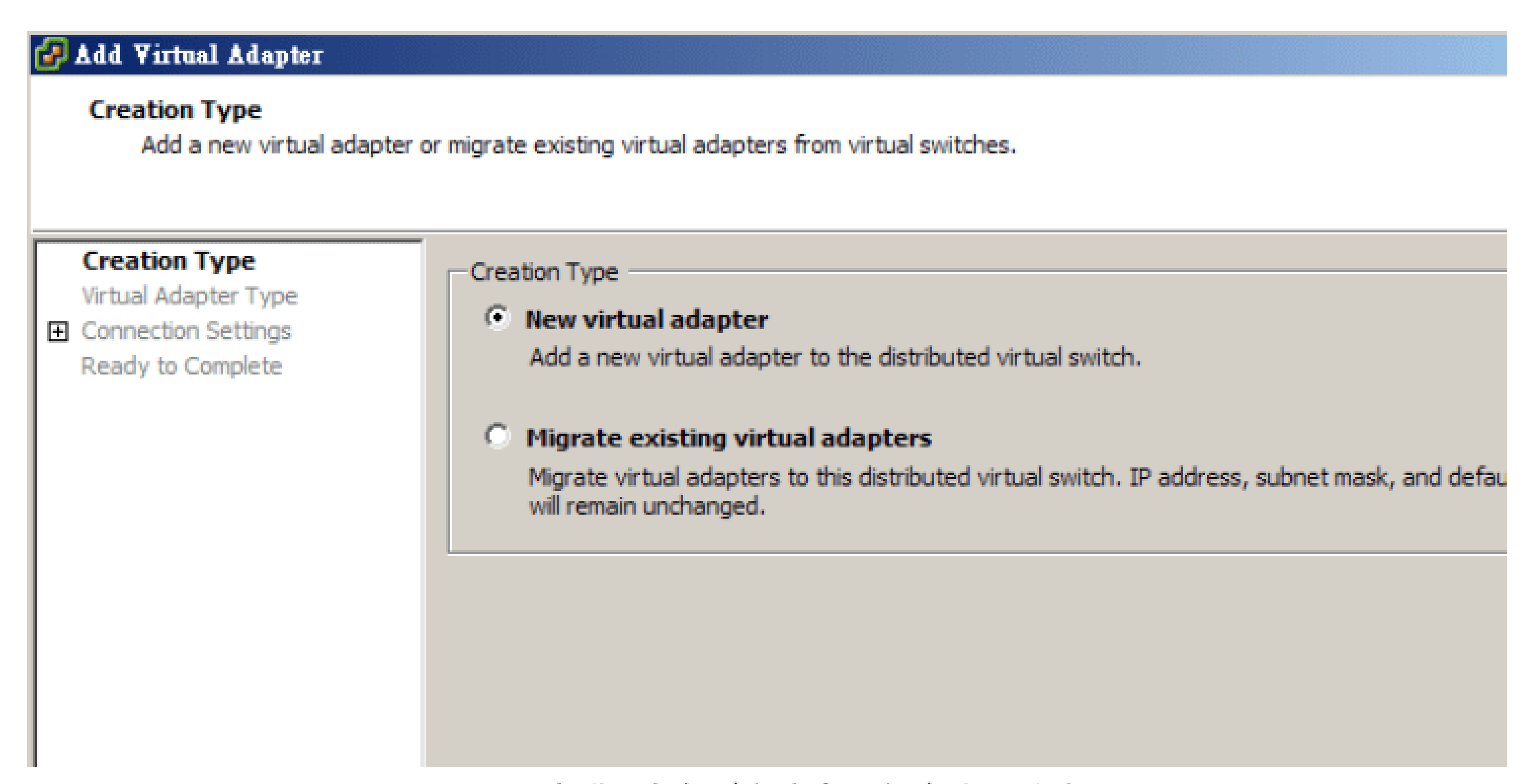
▲ 这次选择虚拟网卡

2. 此时会弹出加入服务的窗口。你可以从中选择 Service Console 或是 VMkernel。之后单击 Add 按钮。



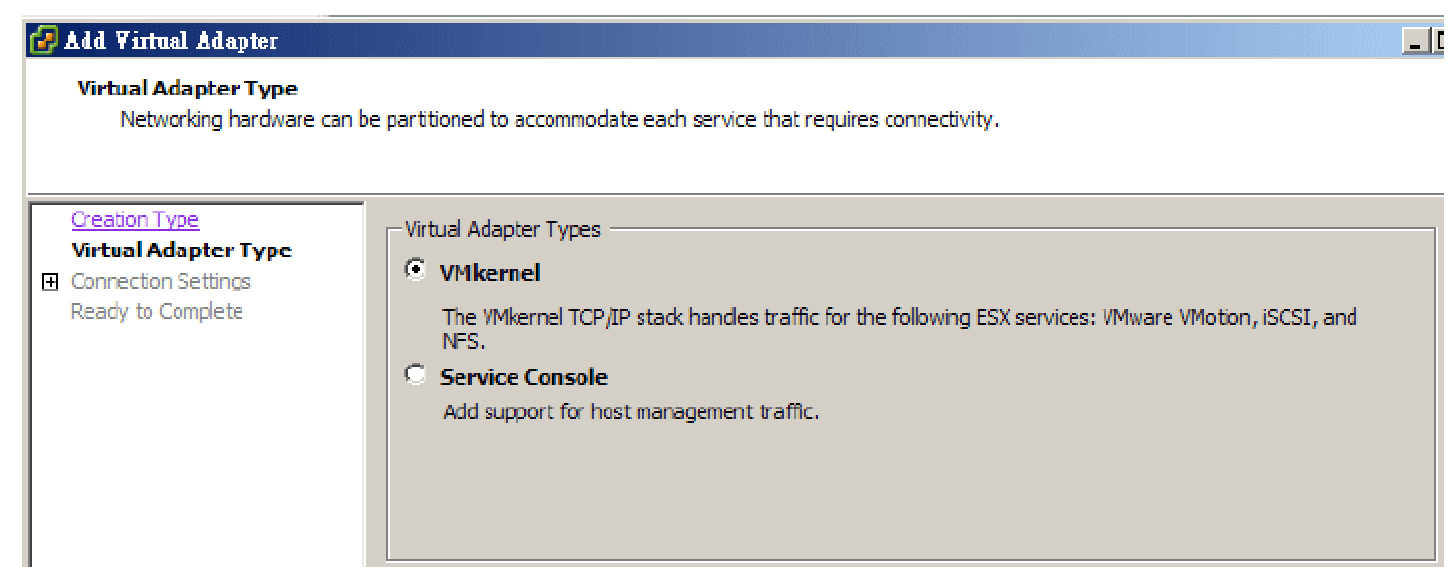
▲ 这里是加入服务的地方

3. 接下来系统会询问你要加入新的虚拟网卡，还是移植当前使用的 VSS。我们就先新建新的虚拟网卡。单击 Next 按钮继续。



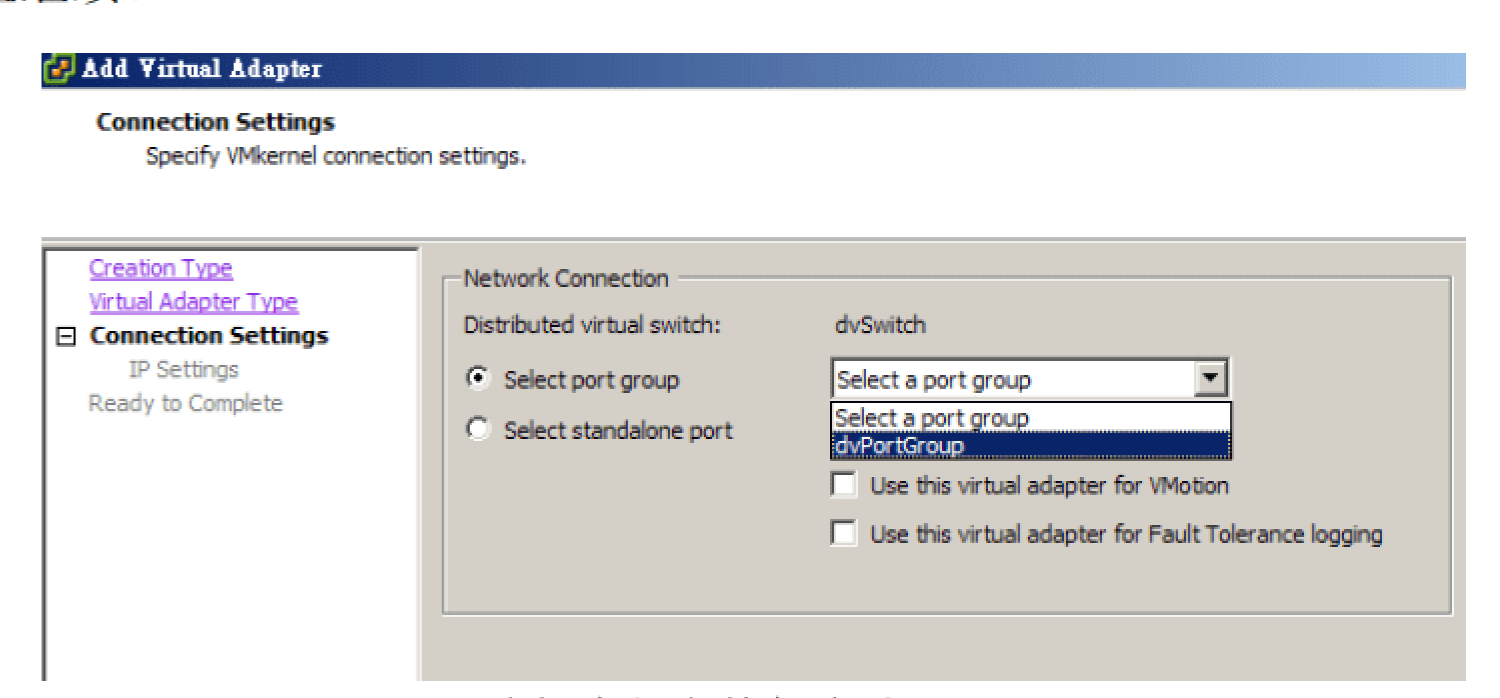
▲ 我们选择创建新的虚拟网卡

4. 接下来就和新 VSS 中的服务完全一样了。选择是 VMkernel 还是 Service Console。我们就以 VMkernel 为例子。单击 Next 按钮继续。



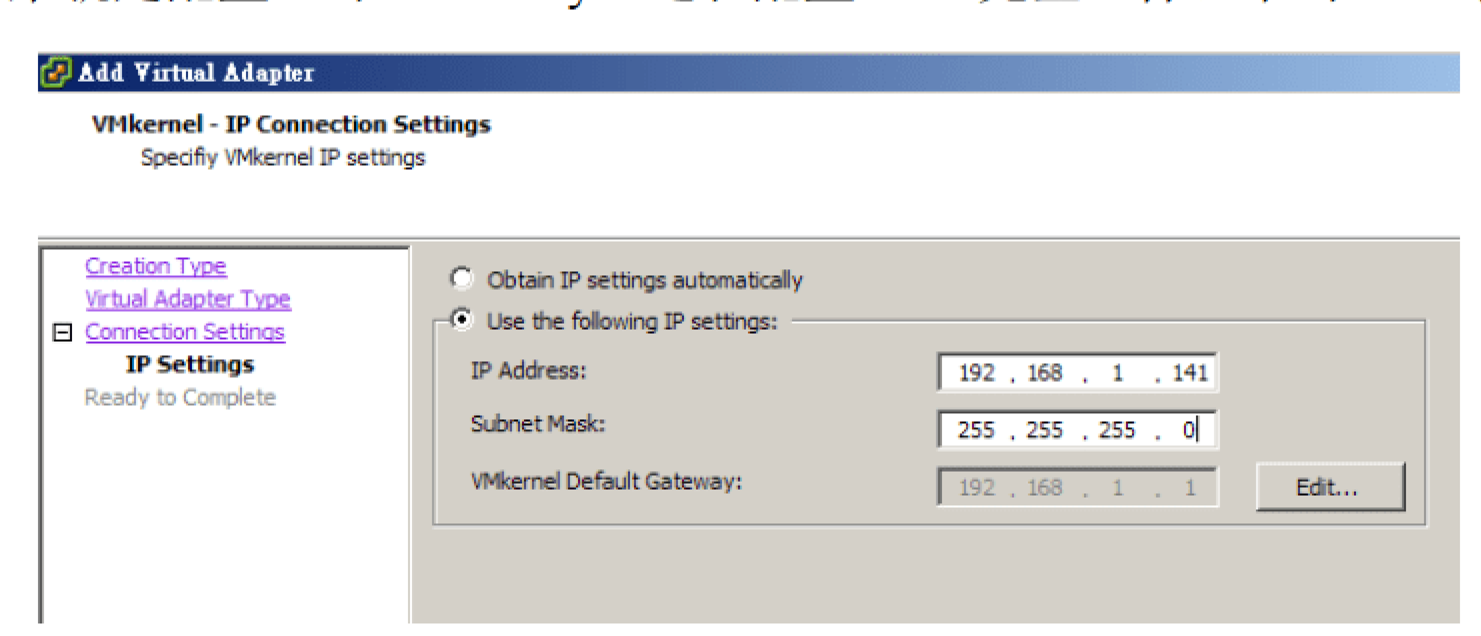
▲ 选择服务类型

5. 我们可以选择 port group 或是选择单独的通信端口。由于我们没有配置单独的通信端口，因此就选择 port group。当然你也要选择是否要用于 VMotion 还是 FT。单击 Next 按钮继续。



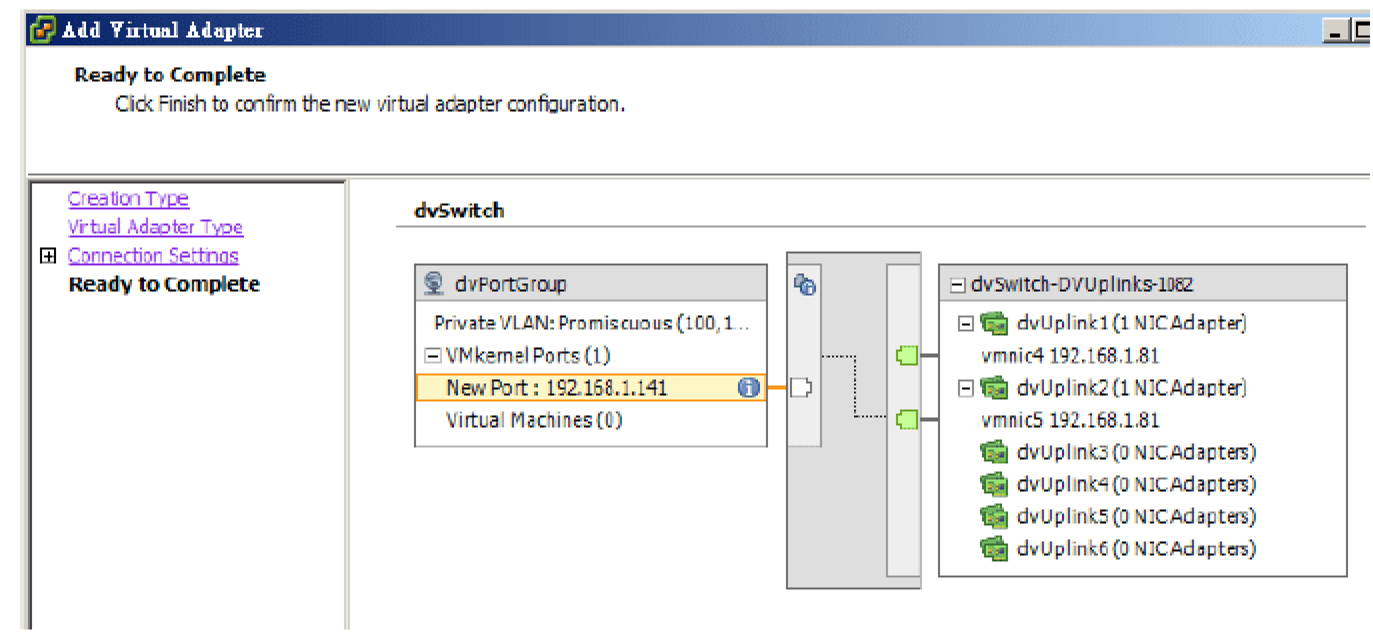
▲ 选择虚拟交换机上的 PortGroup

6. 接下来就是配置 IP 和 Gateway，这和配置 VSS 完全一样。单击 Next 按钮继续。



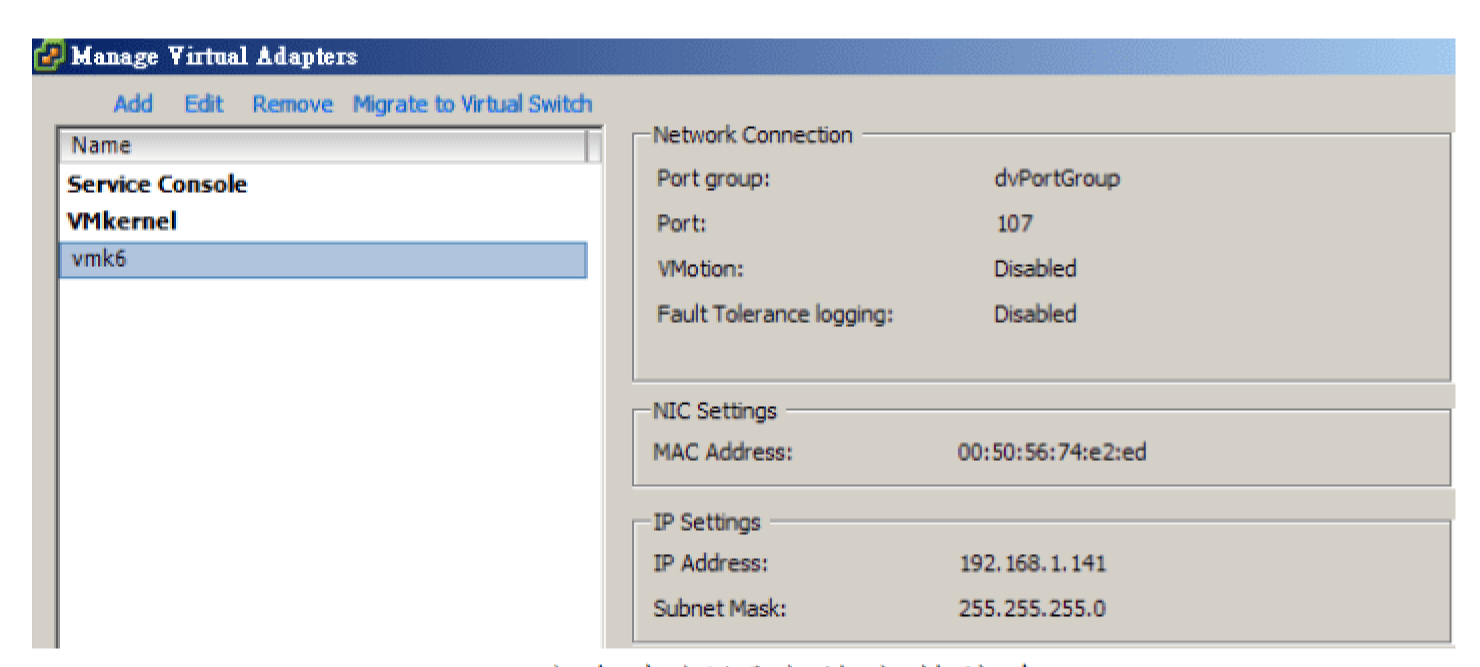
▲ 这里和配置 VSS 的方式完全一样

7. 接下来就弹出当前的连接方式，单击 Finish 按钮落实。



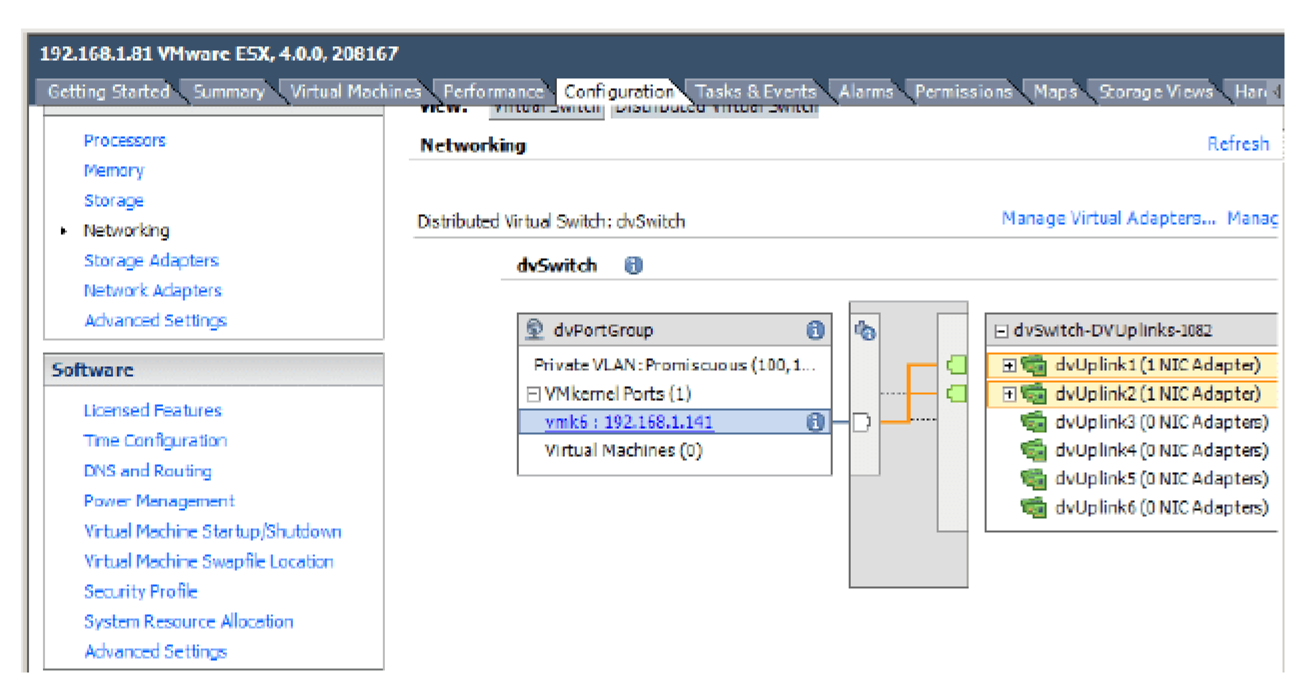
▲ 当前的连接方式

8. 此时我们可以看到 vm6 这个虚拟网卡已经配置落实，并且有完整的信息，如 IP 及激活的功能（如 VMotion）等。单击 Close 按钮结束。



▲ 这个虚拟网卡的完整信息

9. 此时我们可以看到在主画面已经有这个虚拟的网卡，提供了 VMkernel 功能了。



▲ 加入成功

10. 从网络上的任何一个主机也可以 ping 到这个 IP。

```
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 1ms, Average = 1ms
Control-C
^C
C:\>ping 192.168.1.141 -t

Pinging 192.168.1.141 with 32 bytes of data:

Reply from 192.168.1.141: bytes=32 time<1ms TTL=64
Reply from 192.168.1.141: bytes=32 time<1ms TTL=64
Reply from 192.168.1.141: bytes=32 time<1ms TTL=64
Reply from 192.168.1.141: bytes=32 time<1ms TTL=64
Reply from 192.168.1.141: bytes=32 time<1ms TTL=64
Reply from 192.168.1.141: bytes=32 time<1ms TTL=64
Reply from 192.168.1.141: bytes=32 time<1ms TTL=64
Reply from 192.168.1.141: bytes=32 time<1ms TTL=64
Reply from 192.168.1.141: bytes=32 time<1ms TTL=64
Reply from 192.168.1.141: bytes=32 time<1ms TTL=64
Reply from 192.168.1.141: bytes=32 time<1ms TTL=64
Reply from 192.168.1.141: bytes=32 time<1ms TTL=64
Reply from 192.168.1.141: bytes=32 time<1ms TTL=64
Reply from 192.168.1.141: bytes=32 time<1ms TTL=64
Reply from 192.168.1.141: bytes=32 time<1ms TTL=64
Reply from 192.168.1.141: bytes=32 time<1ms TTL=64
```

▲ 也可以从任何同一网段的主机 ping 到这个 IP

> **注意**
>
> **有关 vCenter 的 HA 功能**
>
> 当你使用 vDS 制作新的虚拟网卡功能时，要特别注意在 Cluster 中的 HA 功能改变。由于 HA 使用了 Service Console 来落实 HA 的 Heartbeat，因此在 vDS 中新建了功能之后，HA 的配置可能会出问题，因此一定要特别注意新建的各项 PortGroup。



▲ 在创建了 Service Console 之后，HA 可能会出问题

22.2.3 将 VSS 中的部件移植到 vDS 中

在一个完整的 vSphere 环境中，我们在安装 ESX/ESXi 一开始时，为了方便可能不会配置 vDS，会先配置 VSS 使用。等到将整个 vSphere 环境配置好之后，再将 VSS 中的功能移植到 vDS 中。接下来我们就来看看如何将 VSS 的功能移植到 vDS 中。

1. 移植前的注意事项

当我们要移植现有的 VSS 到 vDS 上时，下面有几个地方要注意，以免移植后整个系统无法使用。

▶ 移植 VSS 到 vDS 的注意事项

（1）在激活 HA 的 Cluster 中，如果将其中一台 ESX/ESXi 的 Service Console 移植上 vDS，在这个 Cluster 中的其他 ESX/ESXi 也得移植，以免 HA 的 Heartbeat 出问题。

（2）移动 VMkernel 的制约较少，但如果在 Cluster 中，移动一个 ESX/ESXi 主机的 VMkernel，一定要移植其他对应主机的 VMkernel。

（3）加入 vCenter 的 Service Console 最好不要移动，以免在引导时无法找到 ESX/ESXi 主机。

（4）如果要使用 VMotion，则加入 vDS 的每一台 ESX/ESXi 主机都必须创建一个 vDS 下的 VMotion PortGroup。使用 VSS 上也行，但会弹出报警窗口。VMotion 将会使用 vDS 和 VSS 的混合模式。

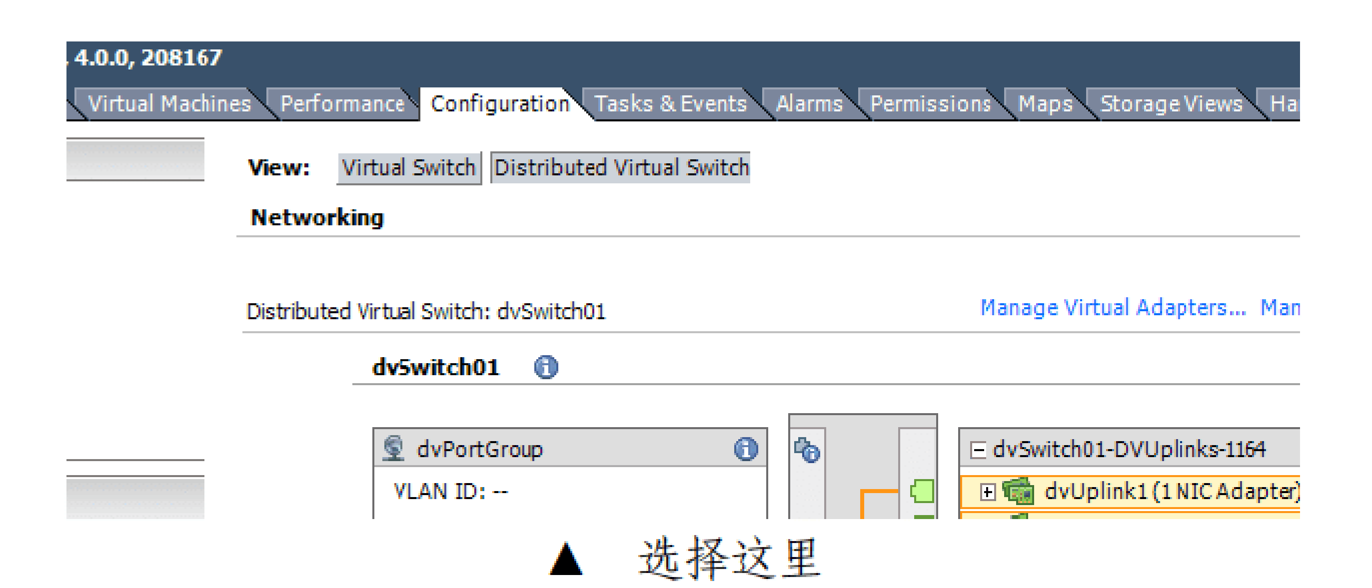
（5）移植过去之后，所有的虚拟网卡还是会维持原来的 IP 及配置。

2. 开始移植 VSS 至 vDS 上

移植 VSS 至 vDS 上，也是使用管理虚拟网卡的方式，我们就来看看完整的步骤。

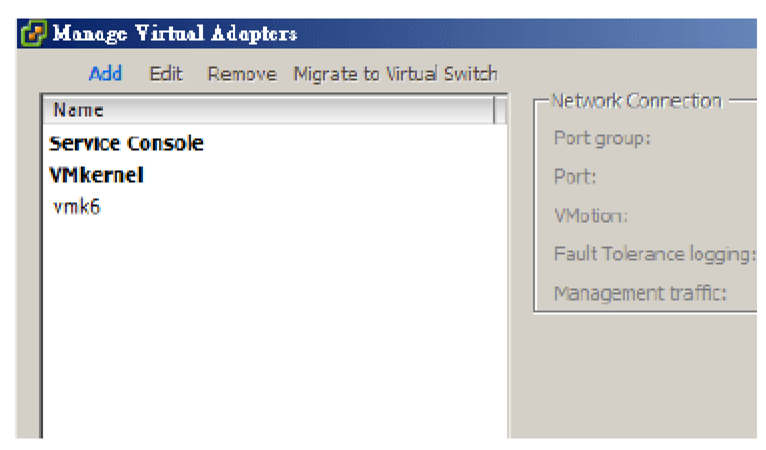
▶ 移植 VSS 至 vDS 上

1. 进入 vCenter 中，并且选择要加入 vDS 的 ESX/ESXi 主机。选择 Configuration/Networking/Distributed Virtual Switch。并且选择 Manage Virtual Adapters。



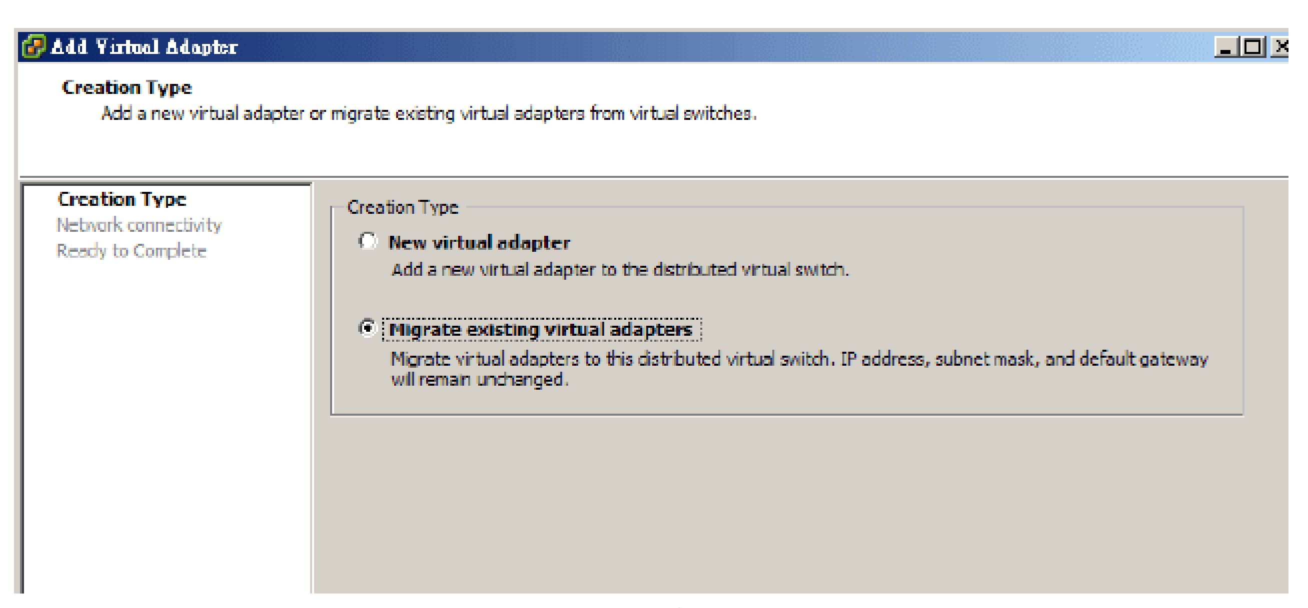
▲ 选择这里

2. 选择 Add 选项。



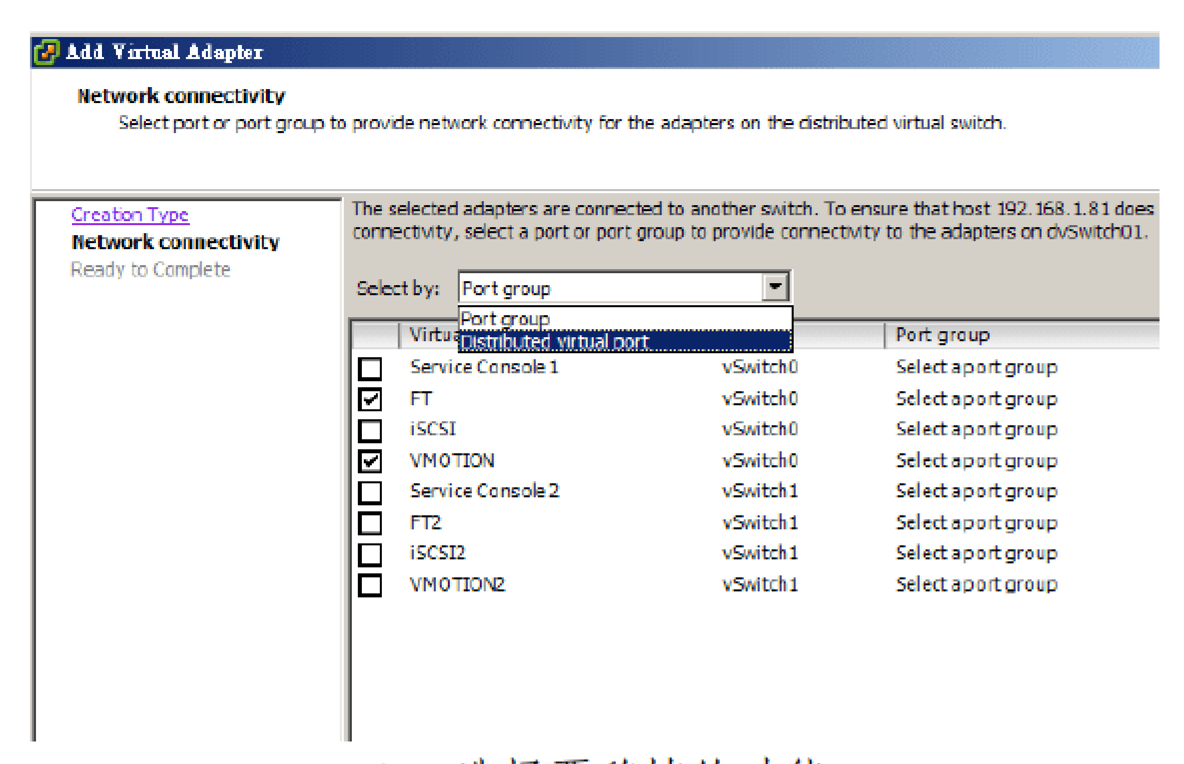
▲ 新建虚拟网卡

3. 选择第二项，就是将当前的虚拟网络功能移植的选项。单击 Next 按钮继续。



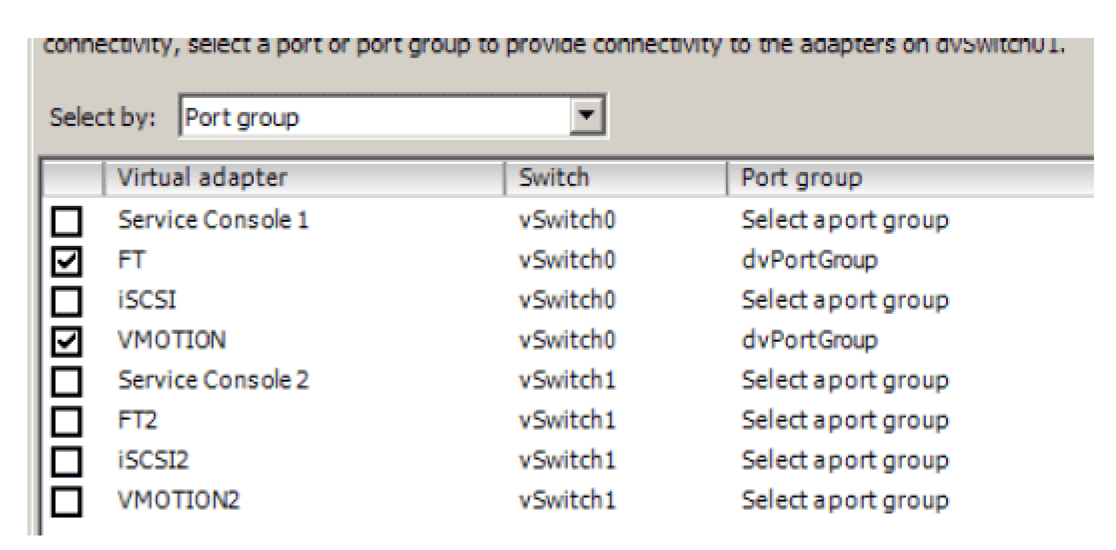
▲ 选择移植而非新建

4. 此时会给出这台 ESX/ESXi 主机所有的网络功能，打勾就是要移植的。我们就将 FT 和 VMOTION 这两个 VMkernel 选中。



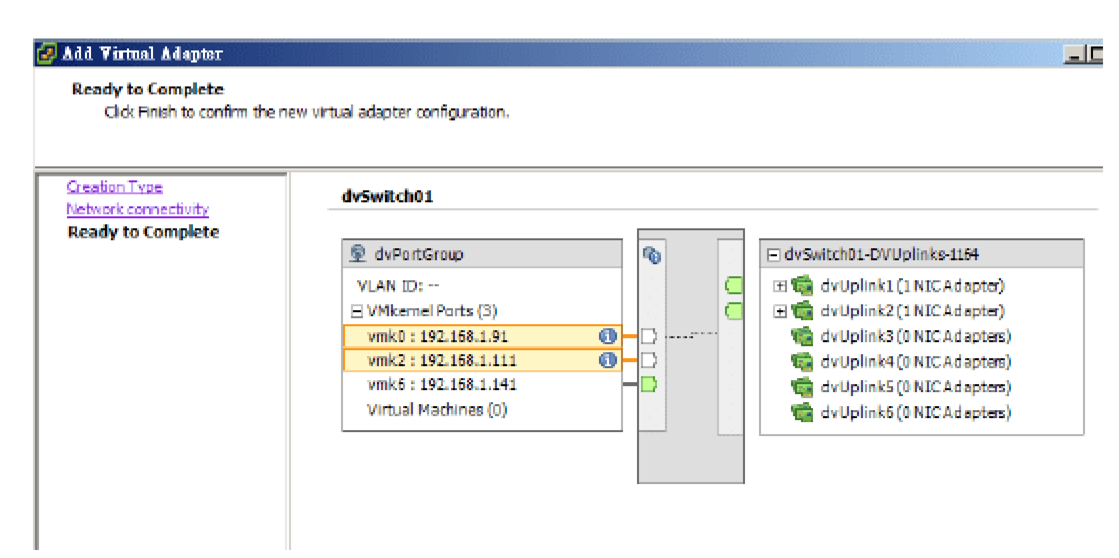
▲ 选择要移植的功能

5. 选完之后，别忘了到后面选择要移植的目的，这里就选择 vDS 中的 PortGroup 即可。单击 Next 按钮继续。



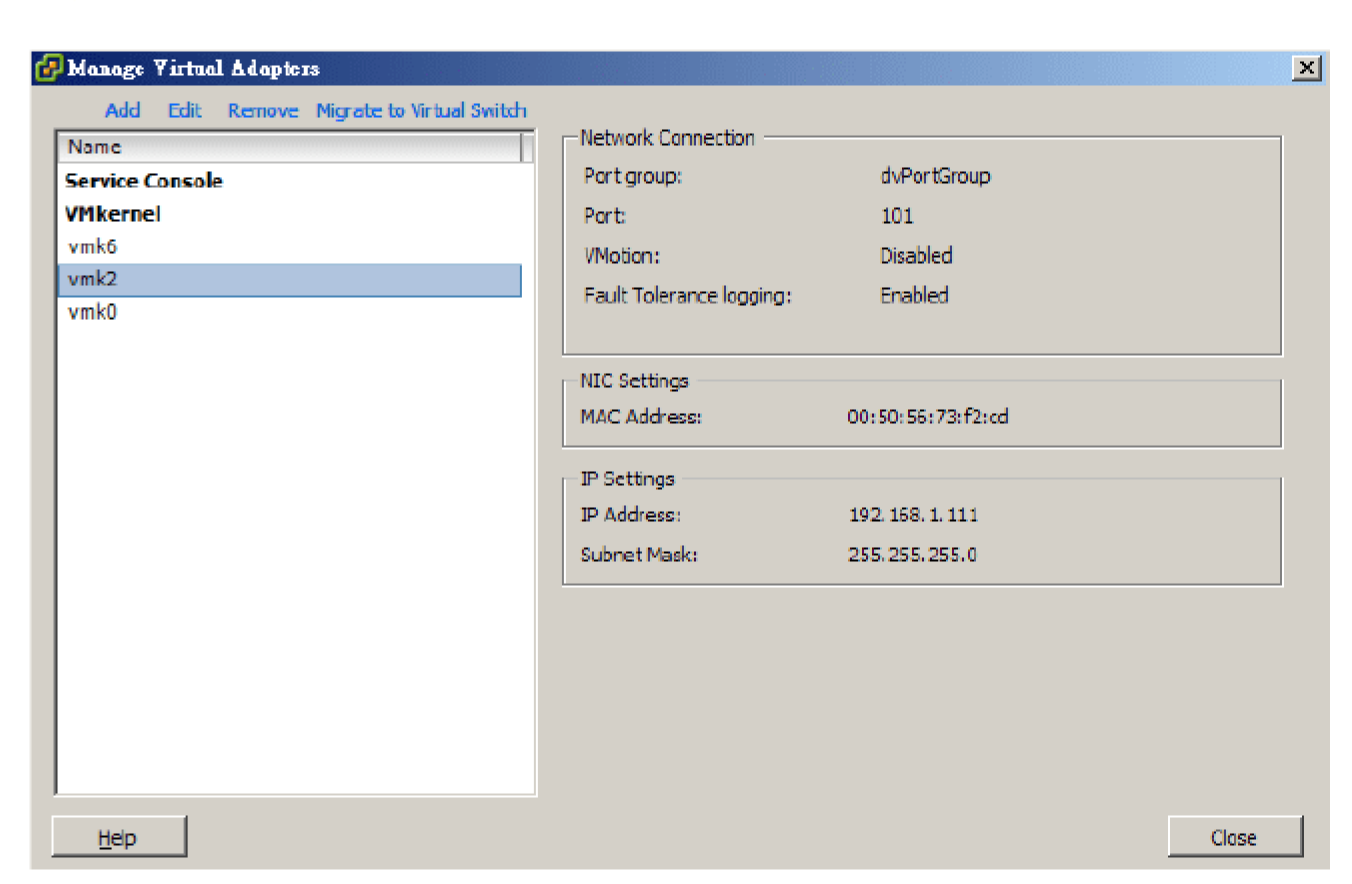
▲ 要移植到什么地方去

6. 接下来会给出移植过去后的连接图。单击 Finish 按钮即可开始移植。



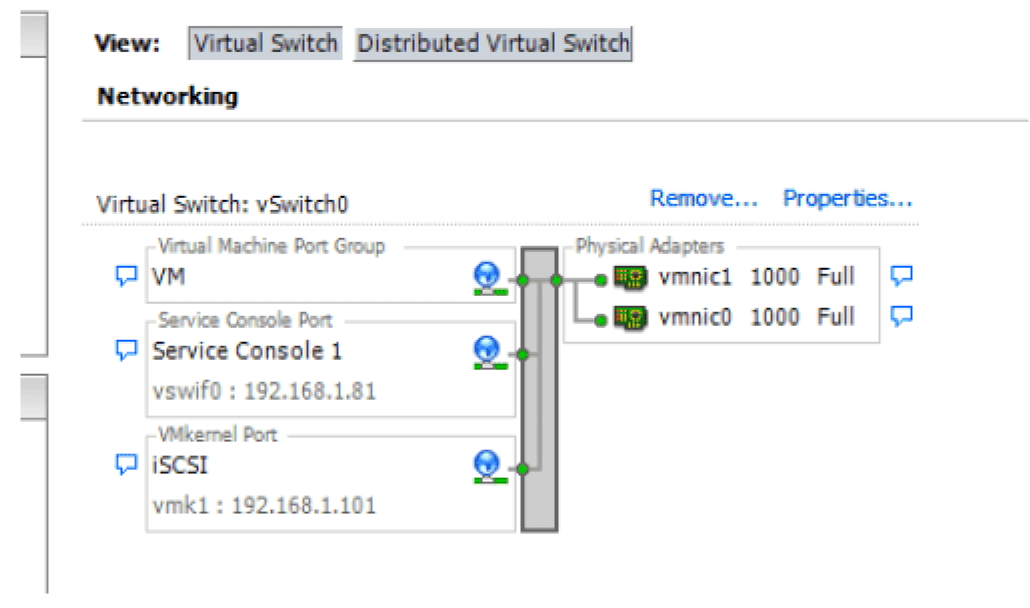
▲ 这是移植后的连接图

7. 我们可以在最后的图中看到移植的几个 VMkernel，并且按下该虚拟网卡可以看到完整的说明。单击 Close 按钮结束。



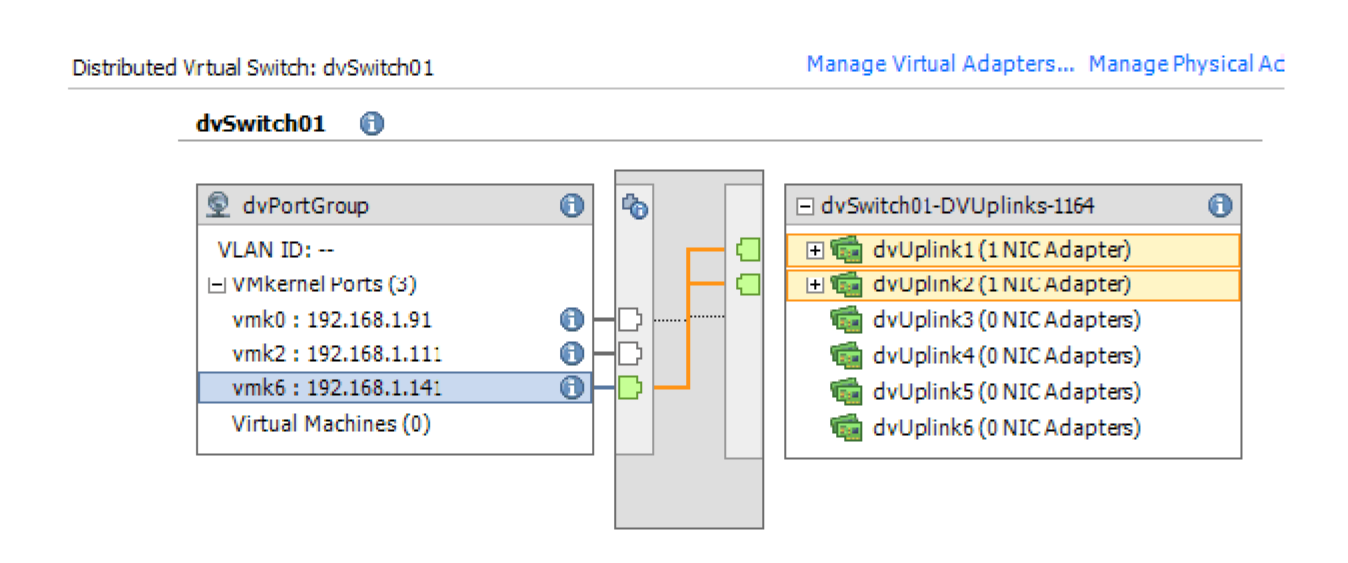
▲ 移植完毕后的图示

8. 在回到网络配置时，我们可以看到 VSS 中的几个 VMkernel 已经不在了。



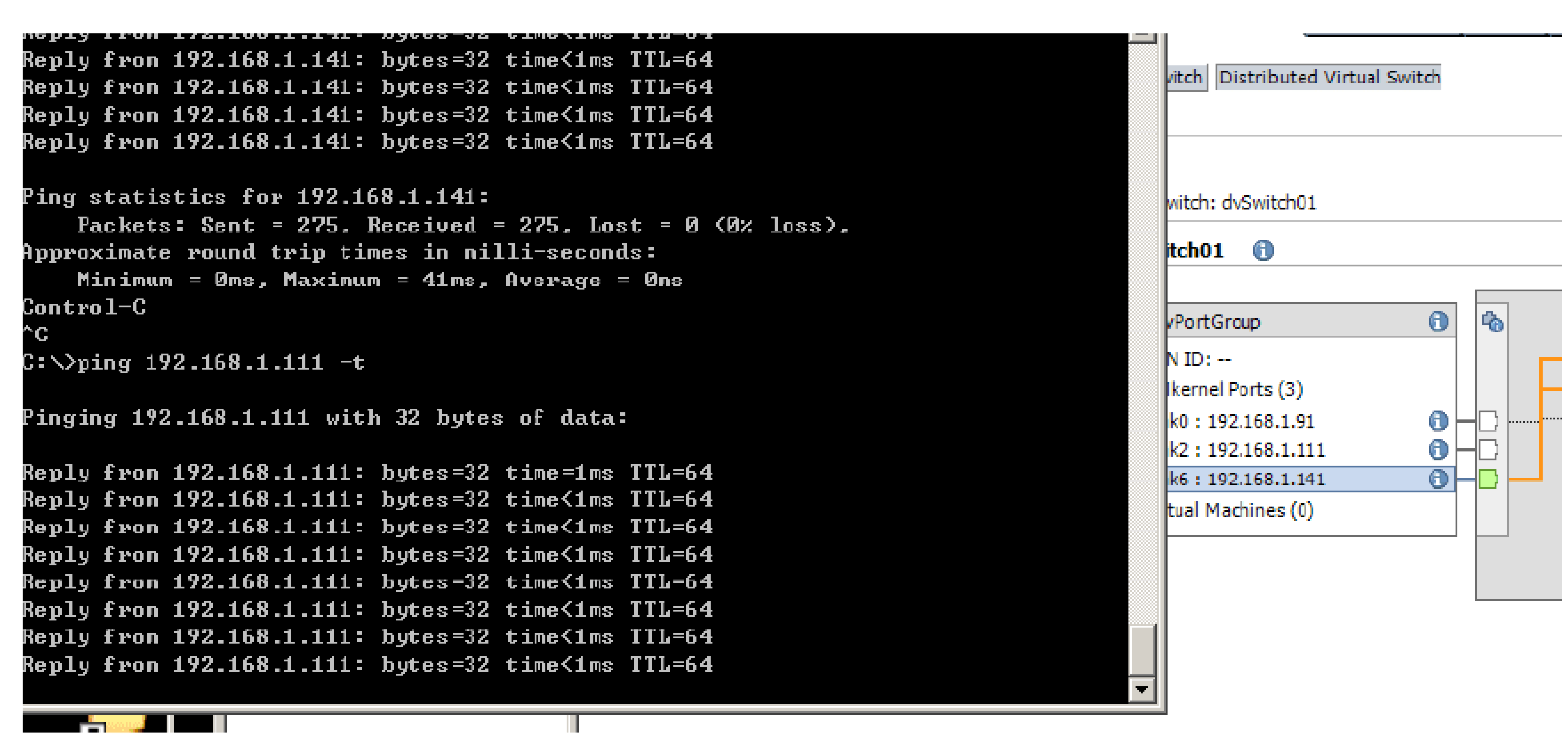
▲ 图示不存在了

9. 可以在 Distributed Virtual Switch 中看到移植过来的 VMkernel。



▲ 已经移植到 vDS

10. 我们使用 Ping 来嗅探移植过去的 VMkernel，发现还能 ping 到。



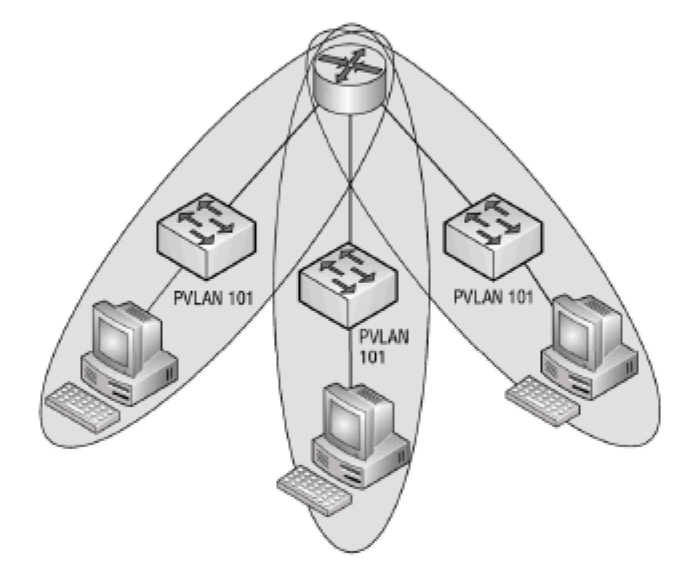
▲ 仍然可以 Ping 到

22.2.4 配置 Private VLAN

Private VLAN（以下简称 PVLAN）是 vSphere 的新功能，也是 vDS 中一个较高级的功能。PVLAN 不能在 VSS 中实现，只能在 vDS 中实现，因此我们在本章的最后，就来看看 PVLAN 的实践。

1. 什么是 PVLAN

PVLAN 的弹出主要是帮助处理多台 ESX/ESXi 主机。基于安全及管理理由，我们常常需要将一些主机放入同一个网段，但又不希望这些主机之间有过多的提交。此时使用 VLAN 虽然可以解决这个问题，但如果主机的数量过多，我们可能需要替每一个主机配置一个 VLAN，这种方式会相当麻烦。如果使用 PVLAN，可以让主机之间的通信端口完全隔绝，但又可以让主机处在同一个网段中，如此一来，不但安全性可以获取保障，管理上又可以用一般网络的方式进行。



▲ 多个 VLAN 之间的主机使用 PVLAN 连接通信端口

2. vSphere 中 PVLAN 的原理

vSphere 中的 PVLAN 使用成对的方式管理，分为主要 VLAN 和次要 VLAN。主要 VLAN 被视为下载用 VLAN。提交至主机的数据使用这个 VLAN。次要 VLAN 则为上传用 VLAN，从主机提交的数据则走这一个 VLAN。

要在 vSphere 中使用 PVLAN，首先你的物理交换机必须支持 VLAN，我们必须先在交换机上配置好 VLAN 的选项，之后再到 vDS 中添加 PVLAN，即可让整个 vSphere 的主机使用 PVLAN。

3. PVLAN 的类型

在 vSphere 中，PVLAN 分成 promiscuous、isolated 和 community 三种类型，我们就分别来看看。

▶ PVLAN 的三种类型

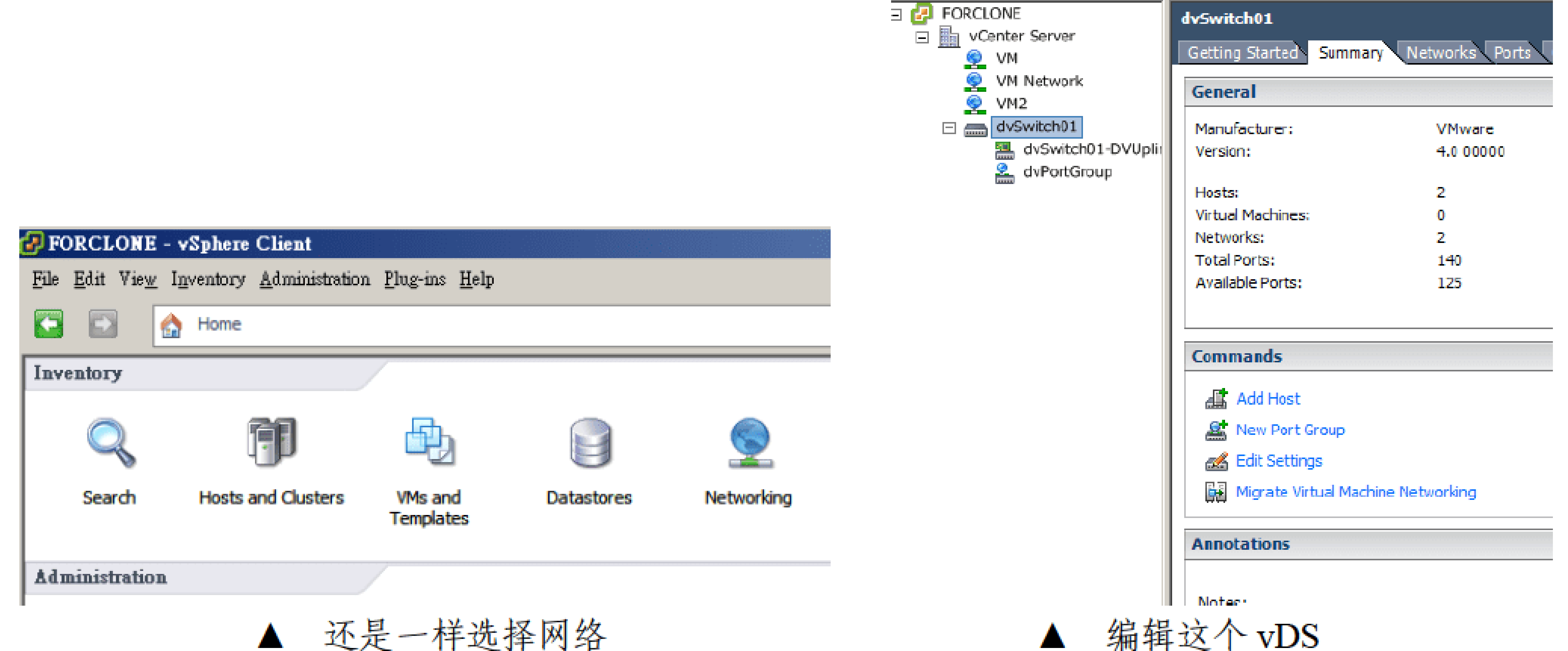
- Promiscuous：这种类型的 VLAN 是允许提交及接受第二的报文至任何 VLAN 的通信端口。一般来说只能使用在主要的 VLAN 中。这一类的通信端口通常给某个 IP 子网的默认网关所保留，如三层路由器。
- Isolated：这一类的 VLAN 在配置时是给次要 VLAN 使用，只允许和同一个次要 VLAN 中 promiscuous 的通信端口沟通。
- Community：可与其他次要 PVLAN 中的 VLAN 沟通，也可以和 Promiscuous 的 VLAN 沟通。

4. 在 vSphere 中配置 PVLAN

在 vSphere 中配置 PVLAN 十分简单，我们只要给定 PVLAN ID，并且给定不同的 ESX/ESXi 及 VM 使用即可，下面就是步骤。

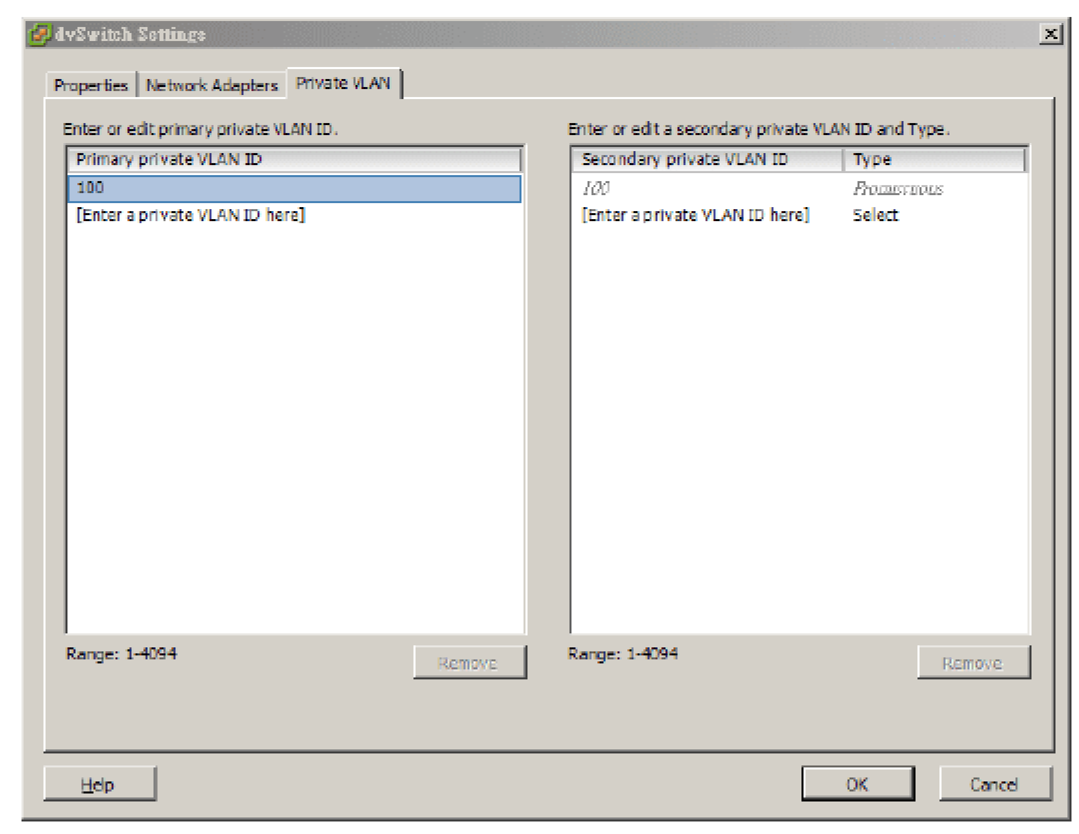
▶ 在 vSphere 中配置 PVLAN

1. 进入 vCenter，并且选择 Home/Networking。
2. 选择要使用的 vDS，并且在 Summary 中单击 Edit Settings 链接。

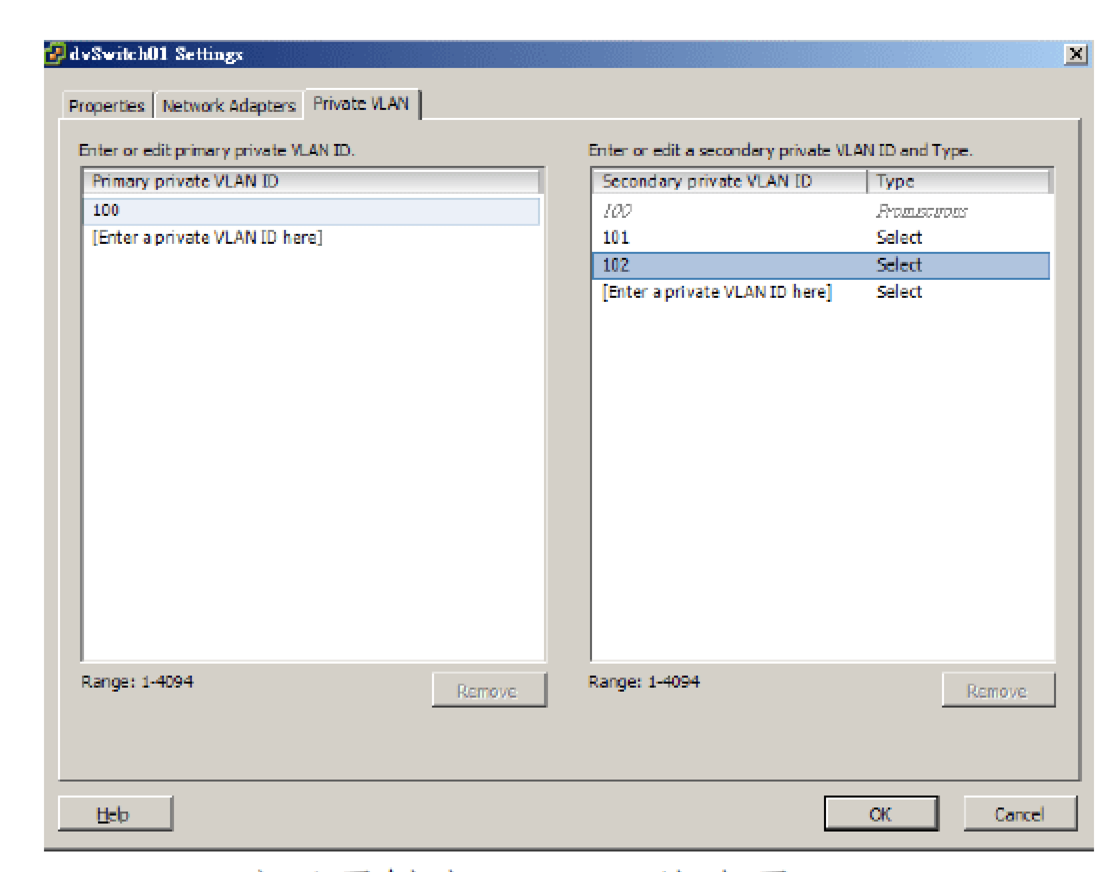


▲ 还是一样选择网络　　　　▲ 编辑这个 vDS

3. 选择 PVLAN 的选项卡，并且在其中键入一个 PVLAN 的编号，如 100。
4. 当我们创建了主要的 PVLAN 编号之后，在图的右边必须创建一个对应的次要 PVLAN 编号，我们就键入 101 及 102 两个。

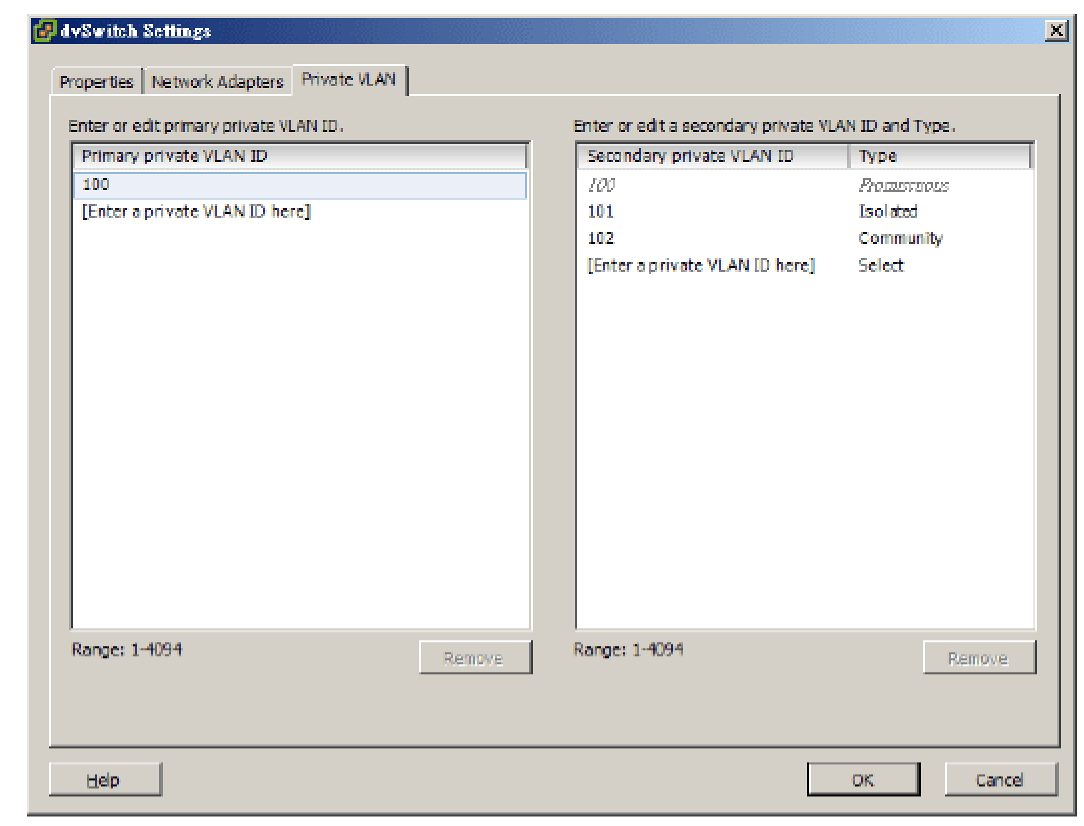


▲ 键入 PVLAN 的主要 VLAN 编号

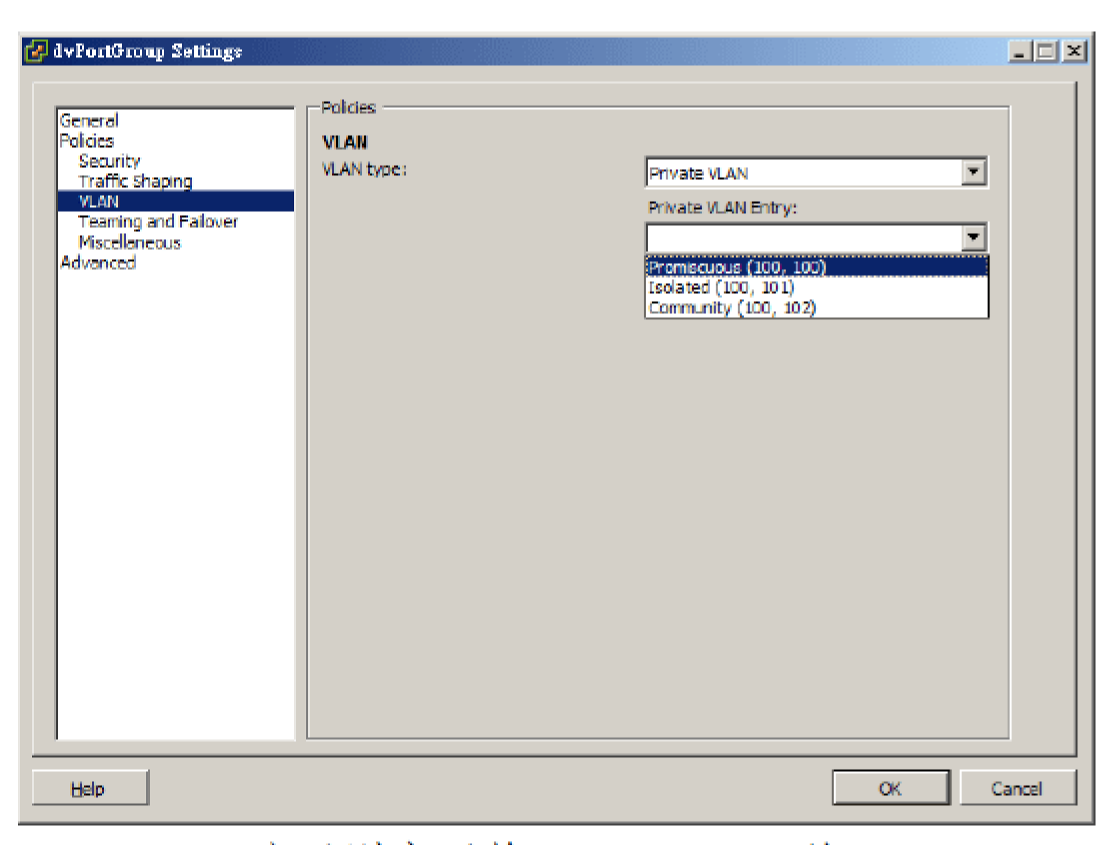


▲ 这里要创建 PVLAN 的次要 VLAN

5. 在创建了次要 PVLAN 编号后，我们要给定这两个 PVLAN 的类型。在同一个 PVLAN 中，只允许一个 isolated，但可以有多个 community。
6. 选择完之后单击 OK 按钮即可。
7. 当配置完 PVLAN 之后，我们就可以配置这个 PVLAN 的 PortGroup。进入 dvSwtich 的 dvPortGroup 中，选择 Edit Settings。



▲ 选择其种类



▲ 这里别忘了将 dvPortGroup 的 VLAN 也换成刚才创建的 PVLAN

我们在 VLAN 的选项卡页中，选择使用 PVLAN，并且选择要走的 VLAN，之后所有使用这个 PortGroup 的网络通信都会使用这个 PVLAN 了。

结 语

vDS 是 vSphere 最新的功能，第一次使用 VMware 产品的人可能觉得理所当然，但是对从 VI3 走过来的工程师来说，vDS 的弹出正是补足整个 vSphere 网络架构的最后一块拼图。vDS 作为一个跨 ESX/ESXi 的交换机或嫌不足，但第三方厂家产品的可扩展性才是 vDS 架构真正的未来，在这一章理解了 vDS 之后，我们将在下一章说明 vDS 的真正实现，Cisco Nexus 1000V 的使用，让大家理解整个 vSphere 网络架构已绝非一家 VMware 可提供了。

第 23 章 使用 Cisco Nexus 1000V 虚拟网络交换机

关键词：

- 什么是 Cisco Nexus 1000V
- Cisco Nexus 1000V 的架构
- Cisco Nexus 1000V 的功能
- 什么是 VN-Link
- 理解 VSM
- 理解 VEM
- 理解 VLAN
- 和 vCenter 创建连接
- 和 ESX 建立连接

在理解了 VDS 之后，总觉得 VDS 的功能虽然强大，但还是没有真正物理交换机的功能这么好用。但在 vSphere 之后，物理和虚拟交换机之间的差距越来越小，主要的原因就是交换机厂家，也开始在虚拟环境中布建功能强大的虚拟交换机了。VMware 提出了一个开放的平台，让各式各样的服务厂家将自身的功能内置，或使用 VA 的方式放入 vSphere 中，本章的热点 Cisco Nexus 1000V，就是网络设备龙头老大所开发的全球第一个虚拟交换机，本章内容多半和 NX—OS 相关，读者必须对网络交换有所理解，阅读时才不会吃力。

23.1 理解 Cisco Nexus 1000V

虚拟化的平台绝对不只是 Hypervisor 的厂家的天下，因为在虚拟化环境中，除了最重要的 Hypervisor 之外，还需要存储、服务器以及网络设备。在 VMware 的产品慢慢成为主流之后，各大厂家也推出了对应的产品。网络设备业者近期在虚拟交换器的技术议题上也有所体现，比如和 VMware 长年合作的 Cisco，即是最早有所动作的一家厂家，他们在 2008 年的 9 月份 VMworld 大会当中，展示了一款能部署在 VMware ESX 平台上的虚拟交换器产品，也就是本章的主角，Cisco Nexus 1000V。

23.1.1 什么是 Cisco Nexus 1000V

Cisco 的虚拟交换机技术称为 VN-Link，主要就是和 VMware 创建交换系统。在这个架构之下，除了软件形式的 Nexus 1000V 也会有硬件的相关专案，像是能与 ESX 平台集成运作的物理交换器 Nexus 5000。这个“VN-Link”在整个系统中扮演了重要的角色，那么什么是 VN-Link 呢？

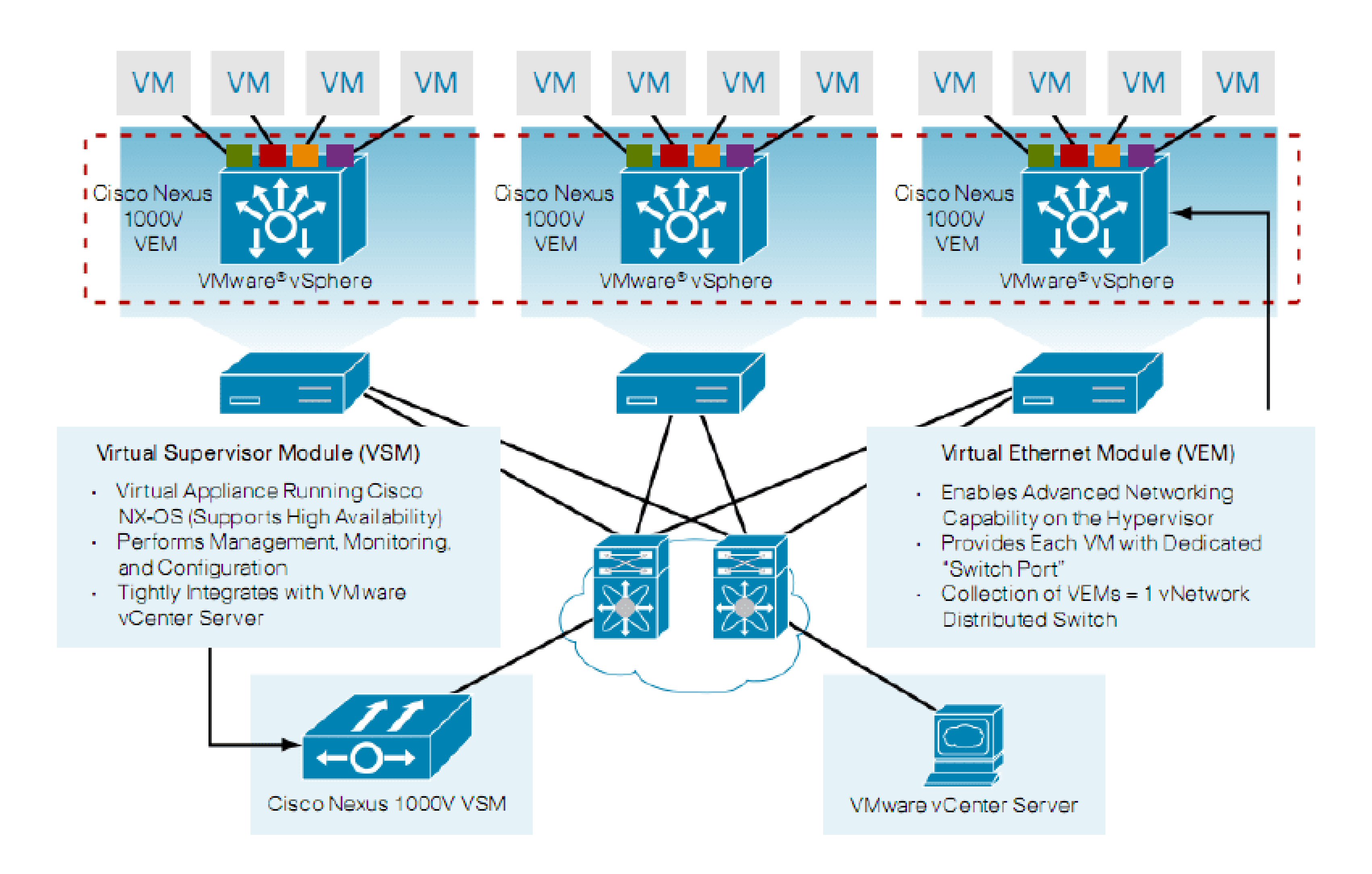
1. 理解 Cisco VN-Link

VN-Link 通过提供管理程序、VM、网络和存储设备间的集成，提高了 VM 的移动性。Cisco VN-Link 提供了与 vCenter 紧密集成、以协作策略的配置模式，采用灵活、统一的管理模式，支持可扩展、移动且高度安全的 VM 安装。

由于服务器虚拟化对服务器和系统管理员来说是一项巨大的挑战。安装服务器虚拟化后，由于每台 ESX 都包含一个或多个虚拟交换机，所有的网络功能都集中到了服务器本身，如此一来，所有的交换机功能就从网络管理员手中转移到了服务器管理员手上，使得服务器管理员需要承担起网络设计、配置、VLAN 创建、应用安全、监控和故障排除等任务，但 IT 管理员不见得熟悉网络配置，因此整个 IT 系统的安全漏洞和问题解决缓慢等风险就添加了。Cisco 能凭借 Cisco VN-Link 来简化这个管理的功能，让你用管理服务器的方式来管理虚拟交换机。

在虚拟化成为了主流之后，网络功能现在都由服务器管理员配置和维护。Cisco VN-Link 的内核是一个称为通信端口简报的 Cisco NX-OS 功能。通信端口简报是一系列网络功能，如 VLAN 分配、服务质量（QoS）参数、访问总控列表（ACL）配置和其他过去应用于特定网络界面的网络功能。与需要分别配置各通信端口的传统网络适配不同，通信端口简报能够在一组通信端口上并发安装。

在一个虚拟环境中，通信端口简报通过 API 集成分发到每个管理程序，vCenter 中作为通信端口群组提供给服务器管理员。在管理新 VM 时，服务器管理员只需使用 vCenter 选择正确的通信端口简报，Cisco VN-Link 就能自动为网络层配置适当的网络连接和策略。



▲ 图为 VN-LINK 的架构（取自 Cisco 网站）

2. 以 VM 为主的管理方式

Cisco VN-Link 提供了实时的网络和安全服务配置，它们在 VM 所处环境的操作始终是一致的。Cisco VN-Link 还采用了以 VM 为中心的管理模式，使服务器管理员在测试新应用时能使用其现有工具和操作任务流程。

Cisco VN-Link 在 VMware 上使用软件虚拟交换机，即 Cisco Nexus 1000V 软件交换机。Cisco Nexus 1000V 交换机采用 Cisco NX-OS 软件，其设计的目的在取代 ESX 的 vSwitch 或是 VDS。Cisco Nexus 1000V 提供了一个与 ESX 完全集成的交换管理功能，并且适用于 Gbps 和 10Gbps 架构，几乎可以完全取代当前使用的硬件交换机。

3. 在多台 ESX 及多个 VM 之间协同作业

通过 Cisco VN-Link 技术，网络管理员能利用可用网络属性定义网络策略，服务器管理员则能为其所安装的应用选择相符的通信端口简报。借此，网络管理员就能在配置和管理网络的并发，允许服务器管理员快捷安装 VM。Cisco VN-Link 使您能定义和安装一致的网络及安全策略，并为您机构中的不同功能组保留原来自我管理的功能。

4. 增强的 VM 安全性

虚拟环境带来了新的安全问题。与硬件交换机相比，虚拟交换机的安全特性相对较为有限。Cisco VN-Link 结合了数据中心级网络安全和运行隔离技术，能够满足当今虚拟服务器环境的安全要求。Cisco Nexus 1000 支持基于角色的访问总控（RBAC）以及握手、授权和审计（AAA）功能。RBAC、AAA 和 VN-Link 的集成，为服务器和网络管理员提供了操作隔离，能在不影响服务器管理员快捷适配 VM 灵活性的情况下实施安全策略。此外 Cisco VN-Link 还支持 ACL 和专用 VLAN，使服务器管理员能对全新的应用集进行虚拟化，而不必为安全考虑而安装专用物理服务器。

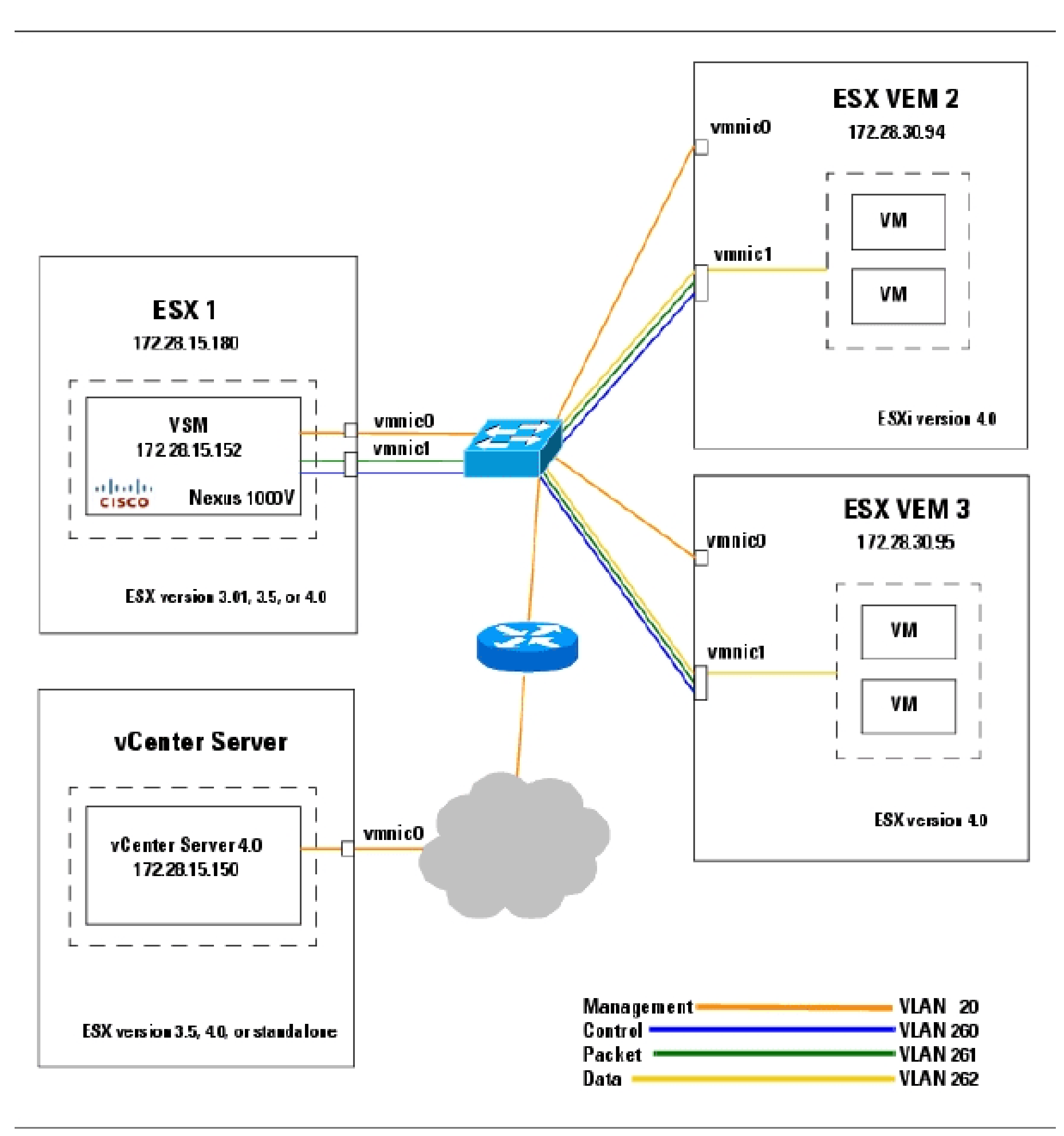
5. 统一的操作和维护界面

Cisco VN-Link，提供了统一的故障排除工具，能快捷诊断并解决服务器虚拟化环境中的网络问题。它们还能使用 Cisco NX-OS 的诊断功能，系统管理员能将一个 VM 的流量对应到位于网络其他

节点的数据报提取器，从而在服务器中详细提取 VM 间的流量，这在物理网络中是无法落实的。

23.1.2 Cisco Nexus 1000V 的架构

Nexus 1000V 虚拟交换机有两个主要部件，即管理 VEM 的外部 VSM 模块（Virtual Supervisor Module, VSM）和每一台 ESX 内部运行的 VEM（Virtual Ethernet Module，VEM）。我们就来看看两个模块的差别。



▲ Nexus 1000V 架构图（取自 Cisco 网站）

1. 什么是 VSM

Nexus 1000V 的 VSM 是一个 VA，主要的功能就是提供 vCenter 一个统一的 VDS。有了 VSM 之后，就可将多个 VEM 作为单个交换机管理。用户不必再安装物理网络线或网卡，VSM 模块支持服务器软件内部运行的 VEM。 配置可通过 VSM 模块进行并自动传播到 VEM。管理员不需要每次在一台主机上的管理程序内部配置软交换机，而是能够在所有由 VSM 模块管理的 VEM 上定义配置并立即使用。

2. 什么是 VEM

Nexus 1000V VEM 是 VMware ESX/ESXi 内核的一部分，能够取代 VMware 虚拟交换机的功能。VEM 充分利用了 Cisco 和 VMware 共同开发的 VMware VDS 的 API，为 VM 提供了进阶的网络功能。Cisco 和 VMware 共同开发这个产品，更确保了 Nexus 1000V 完全实现所有服务器虚拟化的活动，不管是 VMotion，HA，DRS 或是 FT。VEM 从 VSM 模块获取配置信息，也可以运行进阶的网络功能而非虚拟功能，包括 QoS、VLAN、ACL、TrustSec 架构、NetFlow、SPAN、ERSPAN 等。

VEM 具有不间断转发功能，如果与 VSM 模块的连接中断，还能自行通过物理气路转换，能根据最近理解的配置来继续交换流量。由于 Nexus 1000V 是 Cisco 的产品，运行了 NX-OS，因此只要熟悉 Cisco 的工程师就可以操作，因此在 vSphere 弹出之后，大部分的公司已经将网络管理

员和服务器管理员的职位合一了。

23.2 安装及使用 Cisco Nexus 1000V

在理解了 Cisco Nexus 1000V 之后，我们就要在 vCenter 中安装了。事实上，我们可以将 Cisco Nexus 1000V 视为一个第三方厂家提供的 VDS，因此在安装的过程中，只要将 VSM 和 VEM 分别安装即可。

23.2.1 在开始安装之前

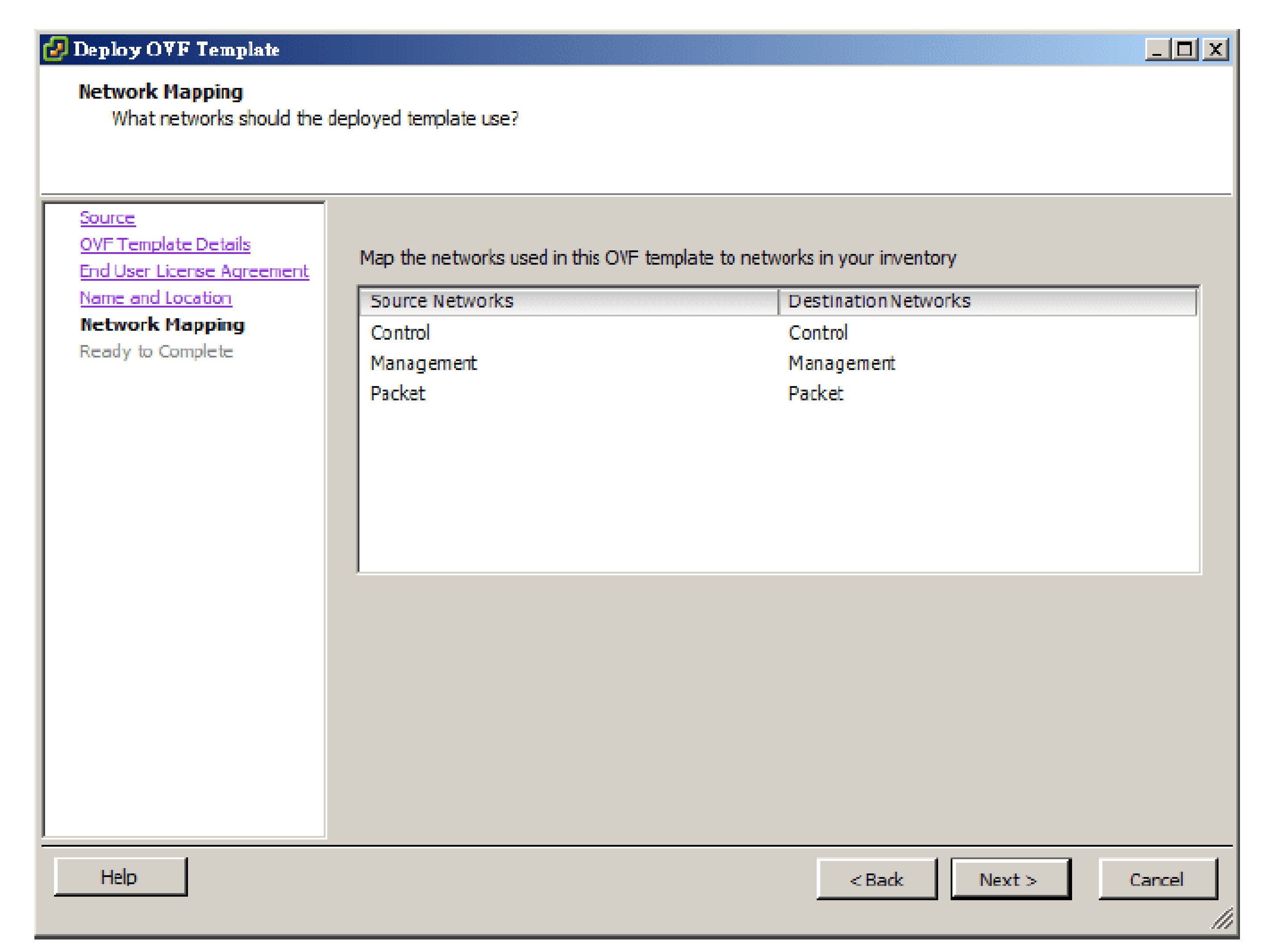
在开始安装之前，我们有几件事要确定，Cisco Nexus 1000V 的安装虽然只是一个 VM，但配置的部分和任何 Cisco 交换机一样使用 NX-OS。NX-OS 是业界公认最复杂的交换操作系统之一，因此我们只会使用最常用的命令，而 Cisco Nexus 1000V 的 VSM 也提供了一个很好用的 Web 界面，简化了安装及配置。

1. 安装前系统必须匹配的条目

在安装 Cisco Nexus 1000V 之前，读者必须先理解整个 Cisco Nexus 1000V 环境的架构，并且你的环境也要匹配下面的配置。

▶ Cisco Nexus 1000V 环境必须匹配的条目

- 一定要安装 vCenter。
- 必须安装 ESX 主机，主要都要拥有 VMware Enterprise Plus 的授权（可使用评估版授权）。
- 运行 VEM 的 ESX 主机一定是要 ESX/ESXi 4.0 以上。
- 至少要一台 ESX/ESXi 主机，如果要运行 VMotion 则是两台以上。
- 至少要有一台 ESX 主机运行 VSM。
- VSM 的 VM 可以在 ESX 主机的 VEM 中运行，该 ESX 主机可以进行管理，VSM 的 VM 也可以在独立的 ESX/ESXi 中运行（3.5 或 4.0 都行），运行标准的 vSwitch。
- 运行 VSM VM 的 ESX 至少要有 4GB 的内存，ESX 主机本身就占了 2GB，剩下的就是运行 vCenter 的功能。
- 每一个主机至少要有下列的几片物理网卡。
- 第一片作为 Service Console/管理。
- 第二片作为 VSM 和 VM 之间的传输，以及 VM 数据的提交。
- 所有 Cisco Nexus 1000V 管理的 ESX 主机都至少能在第二层交换机上连接。
- 如果你使用一组多个交换机，确定交换机之间的 Trunk 连接必须提交相关的 VLAN，包括总控、报文 VLAN。物理的连接必须为一个提交所有配置在该 ESX 主机上 VLAN 的 Trunk 通信端口。
- 在运行 VSM VM 的主机上，总控和报文 VLAN 必须使用标准的 VMware 交换机及 VMNIC。
- 准备好 VEM（vib 格式）和 VSM（ISO 或 OVA 格式）。



▲ 网卡要先给定好不同的 VM Port Group

2. 安装 VSM 的注意事项

在安装 VSM 之前，请确定下面的注意事项。

▶ 安装 VSM 之前的注意事项

- VSM 的 VM 必须和所有管理的 ESX/ESXi 主机有同一网段。
- 运行 VSM 的主机必须是 64 比特的硬件，不管是运行 ESX/ESXi 4.0 或是 3.5。
- 运行 VSM 的 ESX 必须有三个 Port Group，分别是总控 VLAN、管理 VLAN 及报文 VLAN，我们会在下面对应好网卡。

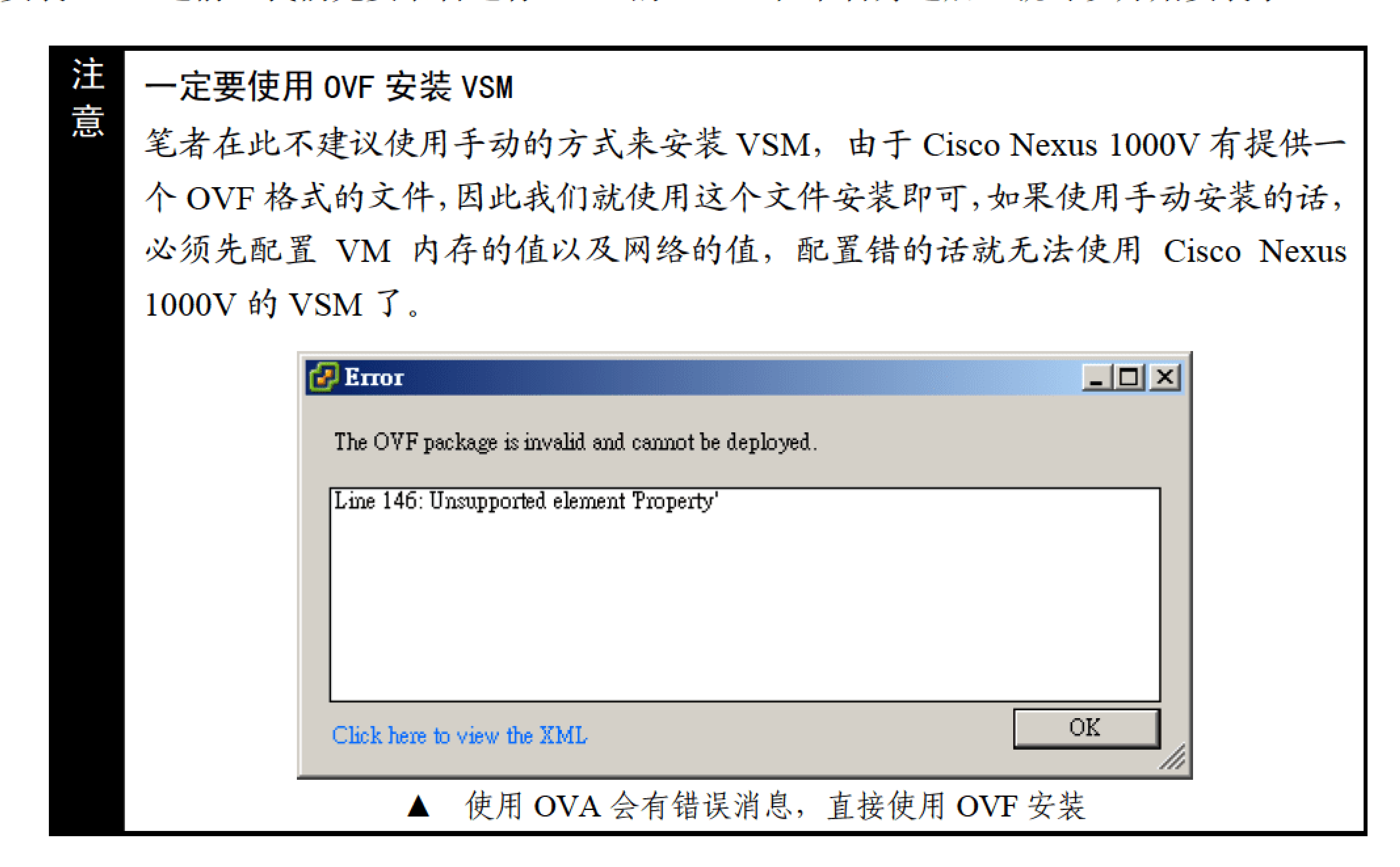
23.2.2 安装 VSM

安装 VSM 之前，我们先要准备运行 VSM 的 ESX。在准备好之后，就可以开始安装了。

注意

一定要使用 OVF 安装 VSM

笔者在此不建议使用手动的方式来安装 VSM，由于 Cisco Nexus 1000V 有提供一个 OVF 格式的文件，因此我们就使用这个文件安装即可，如果使用手动安装的话，必须先配置 VM 内存的值以及网络的值，配置错的话就无法使用 Cisco Nexus 1000V 的 VSM 了。



▲ 使用 OVA 会有错误消息，直接使用 OVF 安装

1. 创建三个 Port Group

在运行 VSM 的 ESX 上，我们必须接上三片虚拟网卡来对应到三个不同的 Port Group。因此我们必须在这台 ESX 上创建三个 Port Group 来进行配置。这三个 Port Group 必须是给 VM 用的，而非 Service Console 或是 VMkernel。有关创建 VM 的 Port Group，读者们可以引用前面有关网络配置的部分，对一章一章读过来的读者来说应该是驾轻就熟了。

表 1　本章使用的网络配置

网卡	VLAN	说明	本章配置的编号
第一片网卡	Control	E1000	260
第二片网卡	Management	E1000	20
第三片网卡	Packet	E1000	261

2. 使用 OVA 安装 VSM

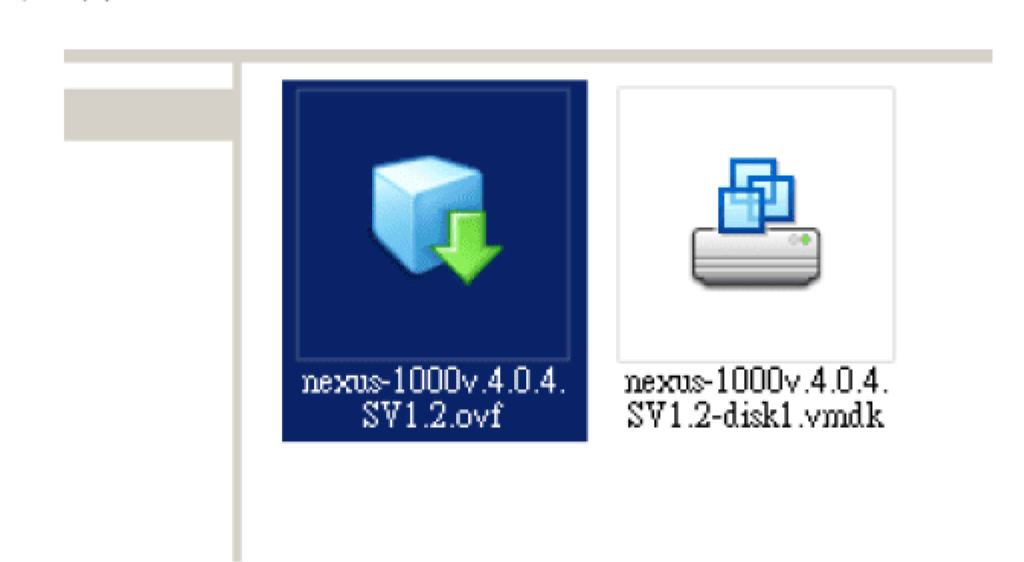
接下来我们使用 OVA 来安装 VSM。下面就是详细的步骤。

▶ 安装 Cisco Nexus 1000V 中的 VSM 模块

1. 首先到 Cisco 网站下载最新版的 Cisco Nexus 1000V4.0（4）SV1（3）。
2. 下载回来之后，将这个压缩文件解开。我们要安装的就是其中\VSM\Install\OVF\nexus-1000v.4.0.4.SV1.3.ovf 这个文件。

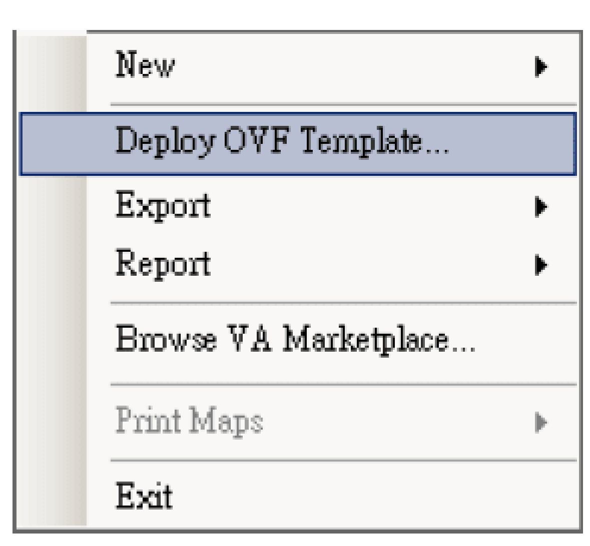


▲ 下载回来的文件



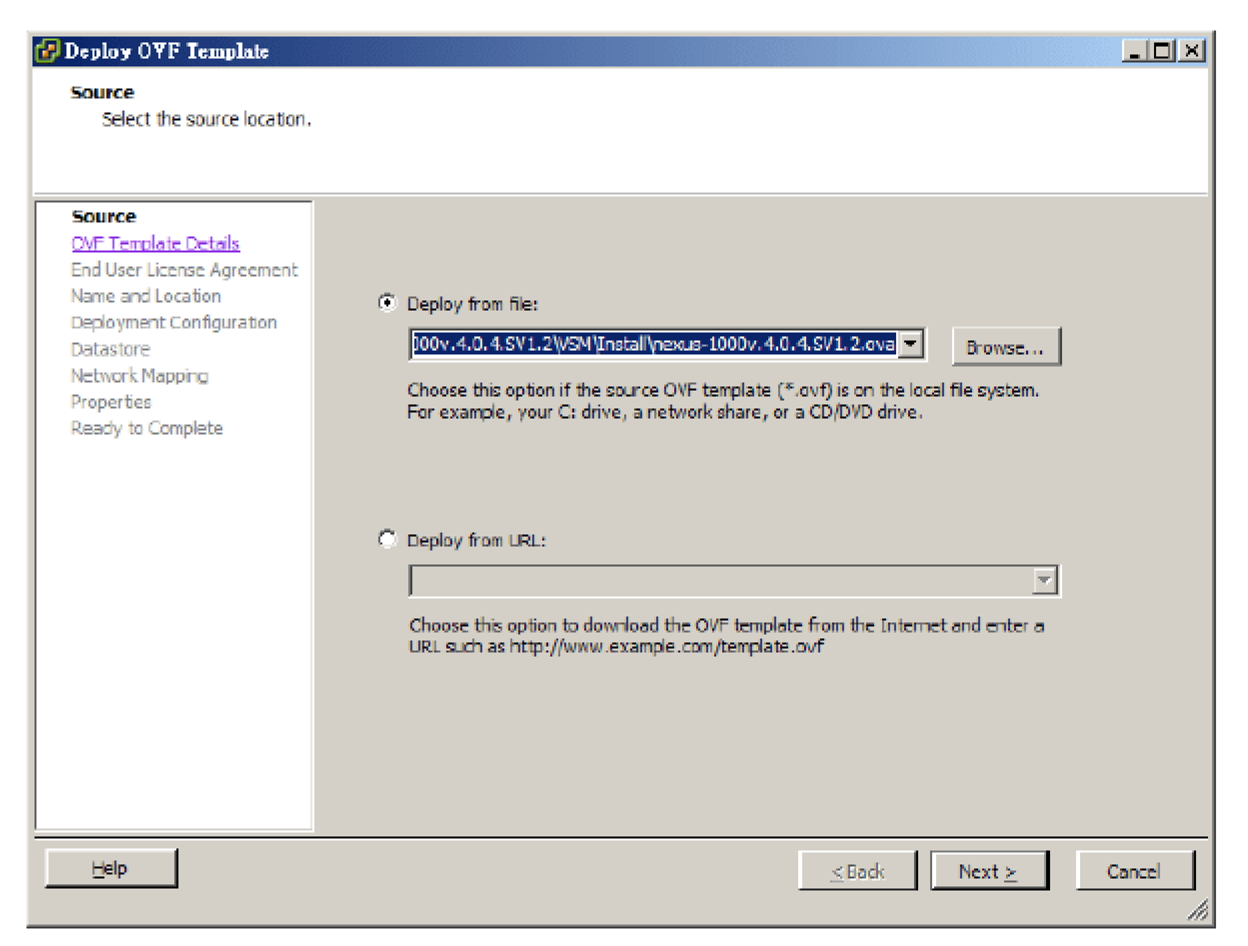
▲ 解开后要使用 OVF

3. 进入你要安装 Cisco Nexus 1000V 的 vCenter。
4. 选择 File/Deploy OVF Template 选项。



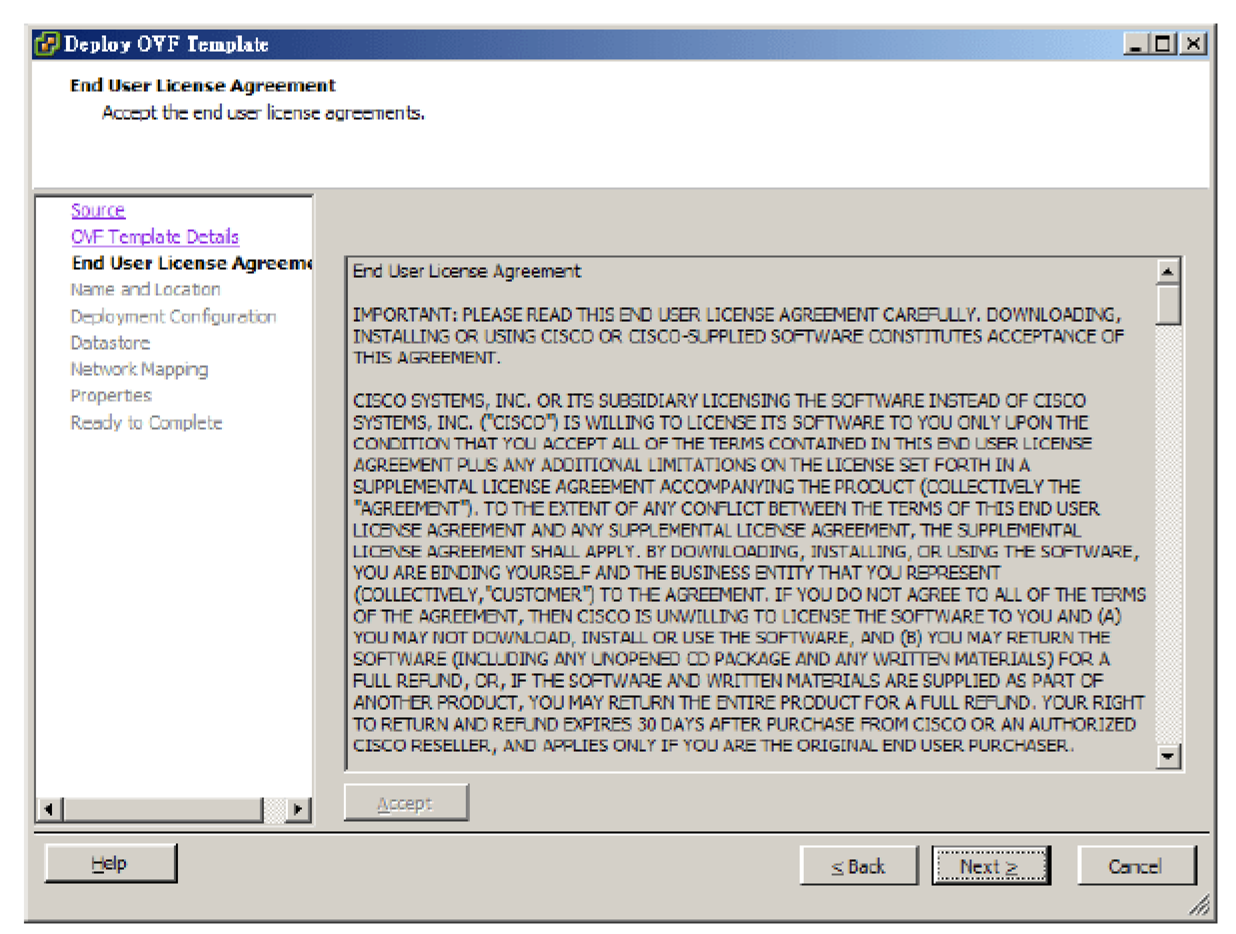
▲ 选择选项

5. 此时会显示当前的这个 OVF 的详细数据，单击 Next 按钮继续。



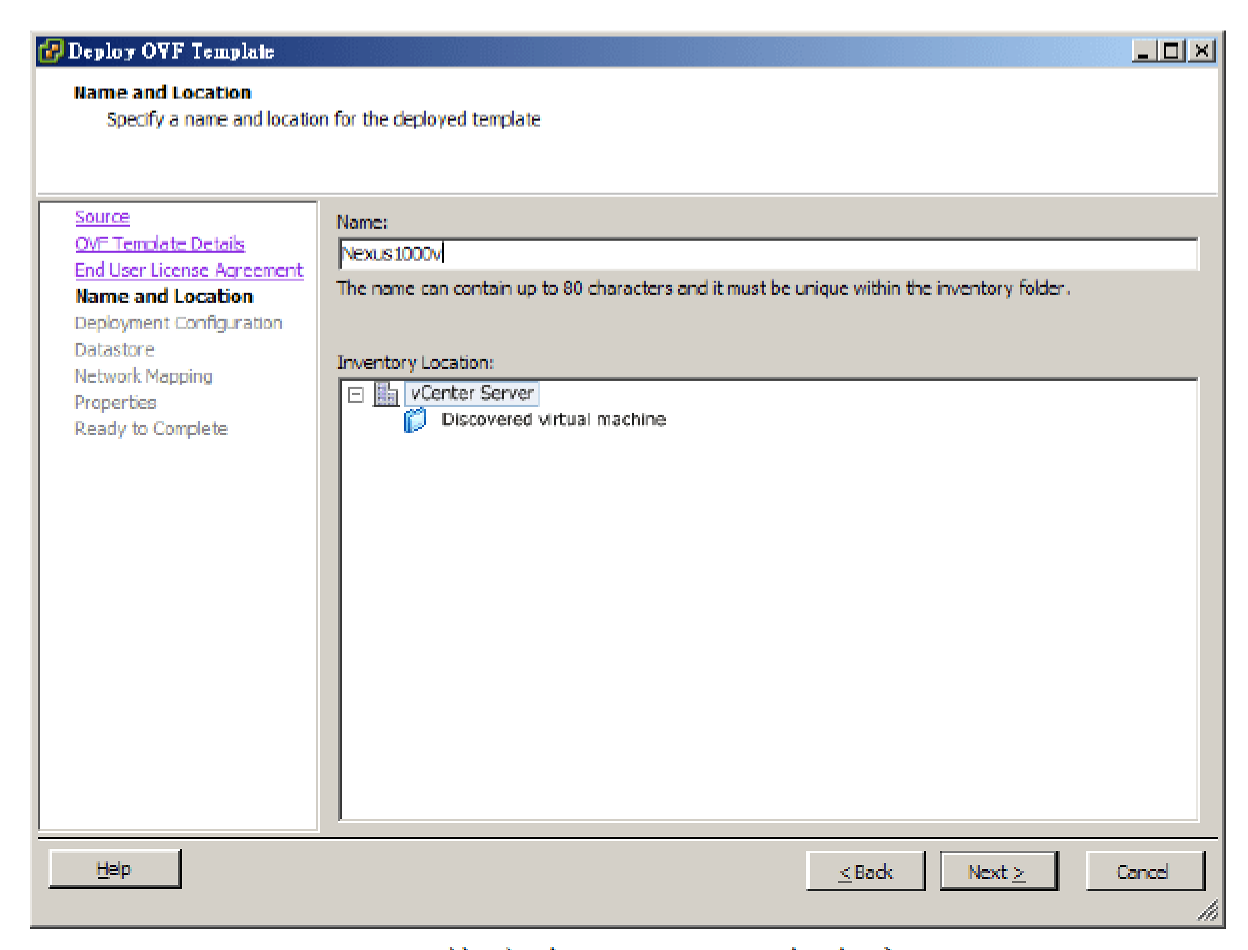
▲ 选择 OVF 来安装

6. 接下来会进入版权画面，单击 Next 按钮继续。



▲ 版权画面

7. 此时替这个 VM 取一个名字，并且选择置放的地点，选择完之后单击 Next 按钮继续。

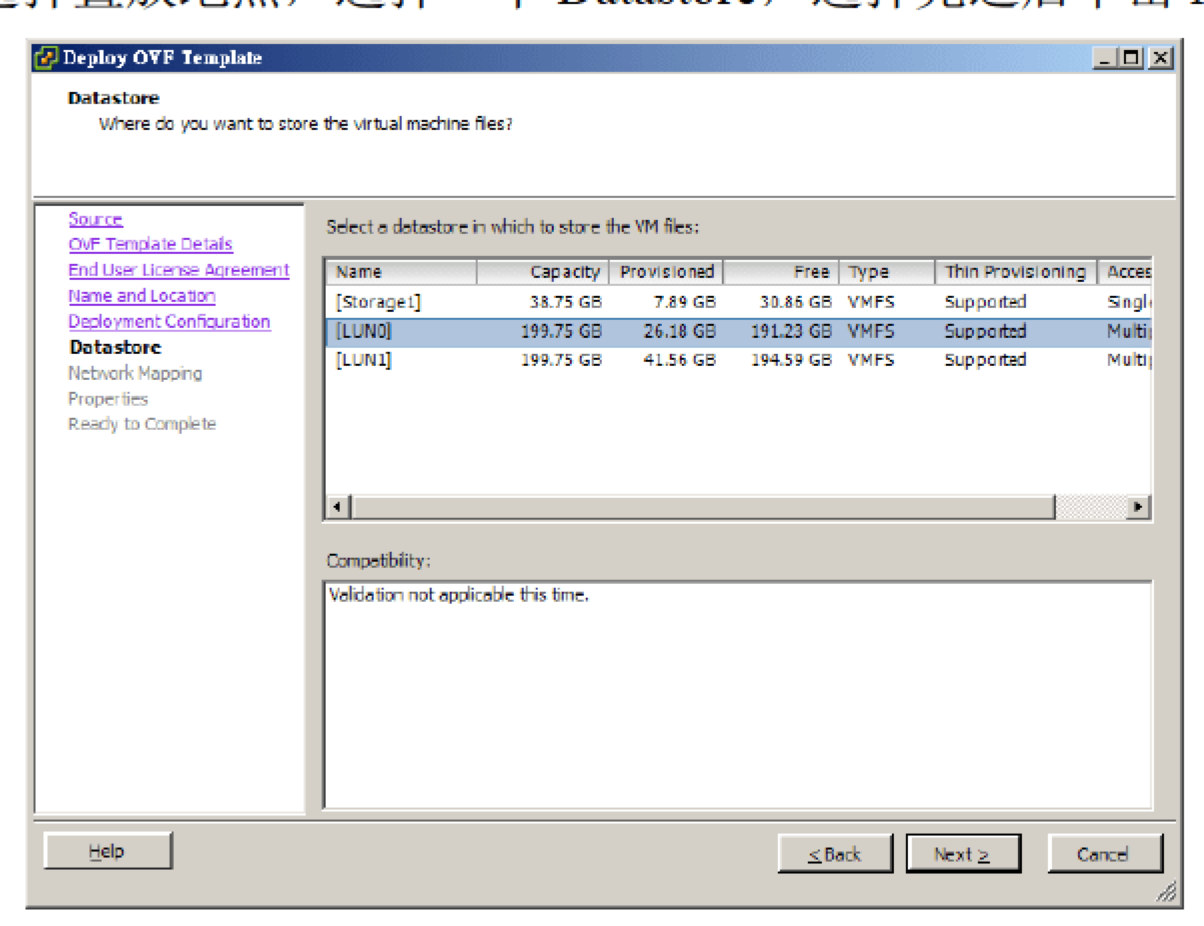


▲ 替这个 VM 取一个名字

8. 接下来选择配置的手段，我们就选择“手动配置”，单击 Next 按钮继续。

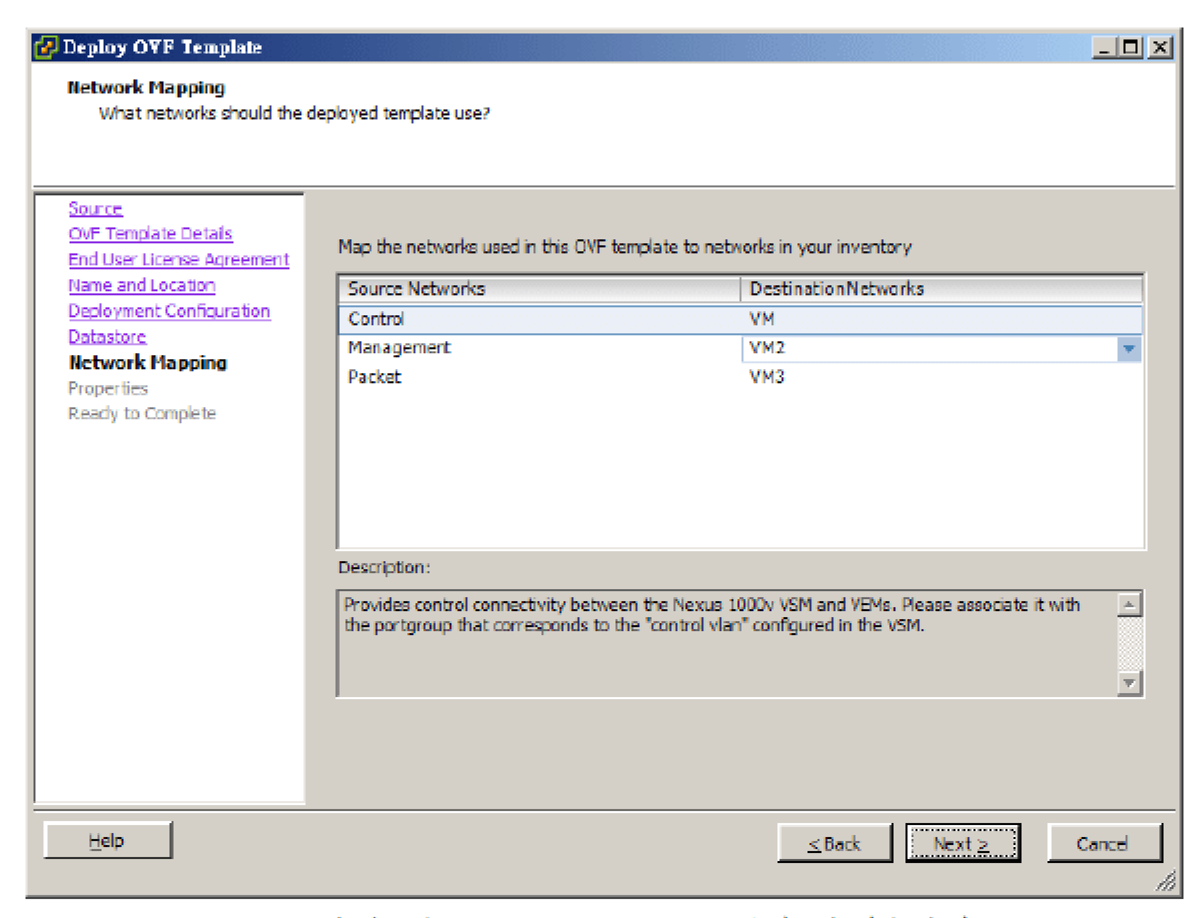
▲ 选择手动配置

9. 接下来会选择置放地点，选择一个 Datastore，选择完之后单击 Next 按钮继续。



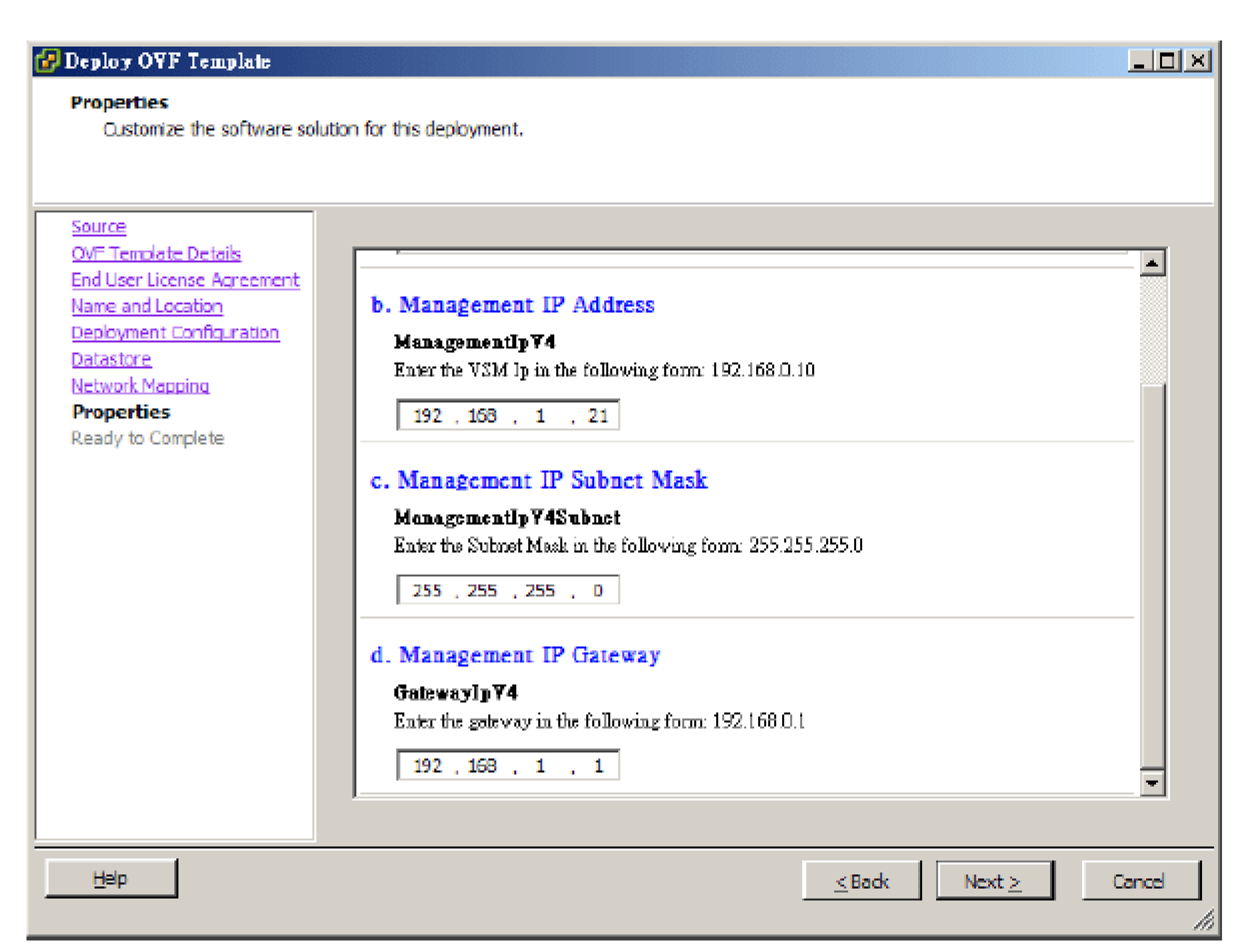
▲ 放置的 Datastore

10. 接下来是选择对应的 VM Port Group。由于之前已经创建了三个 Port Group 对应，因此名字对应完之后单击 Next 按钮继续。



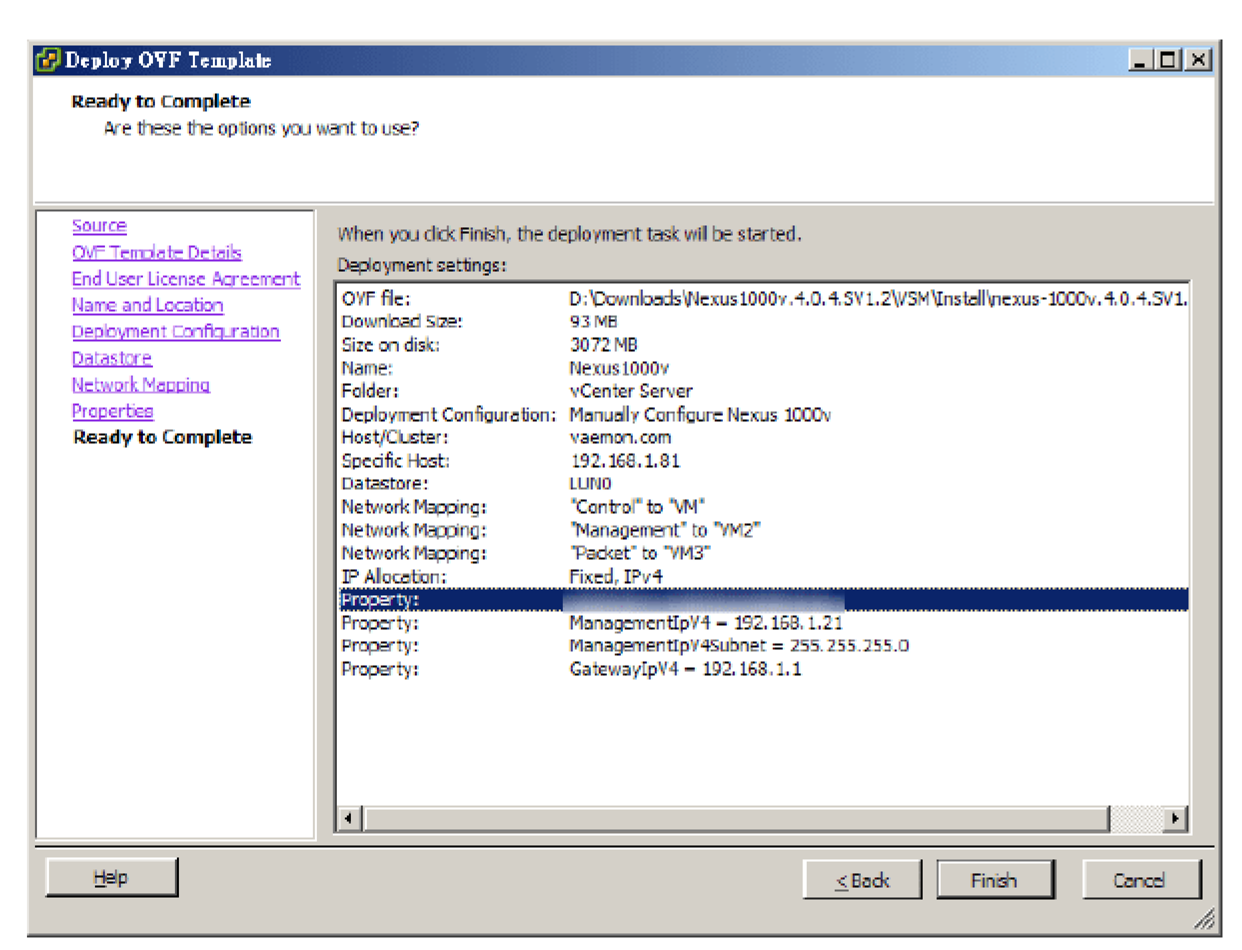
▲ 对应的 Port Group，要事先创建好

11. 接下来是配置其中管理的 IP 位置。这也是之后我们进入 Web 界面管理的 IP。单击 Next 按钮继续。



▲ 键入管理 IP，这个值很重要，要记得

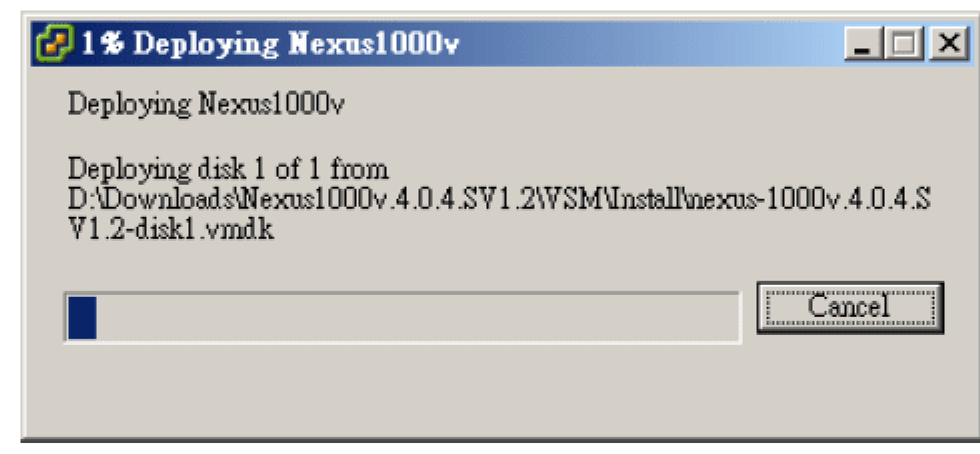
12. 到此会创建一个清单，单击 Finish 按钮则进行安装。



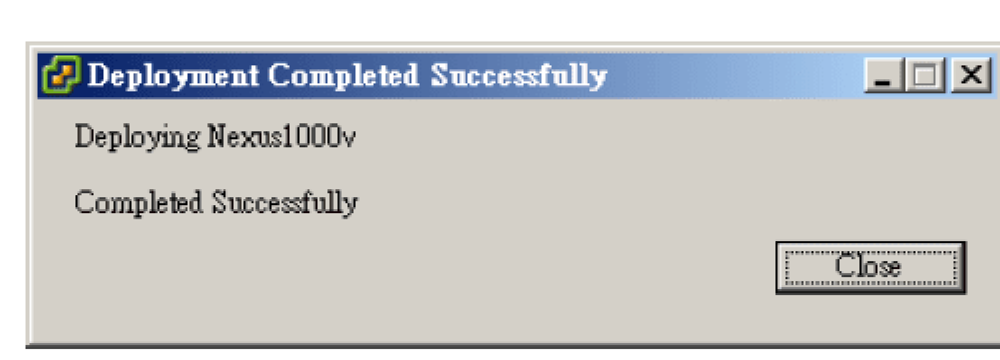
▲ 创建清单

13. 此时系统会进行安装。

14. 当弹出如下图所示的画面时，意味着安装完毕。



▲ 开始进行安装



▲ 安装完毕

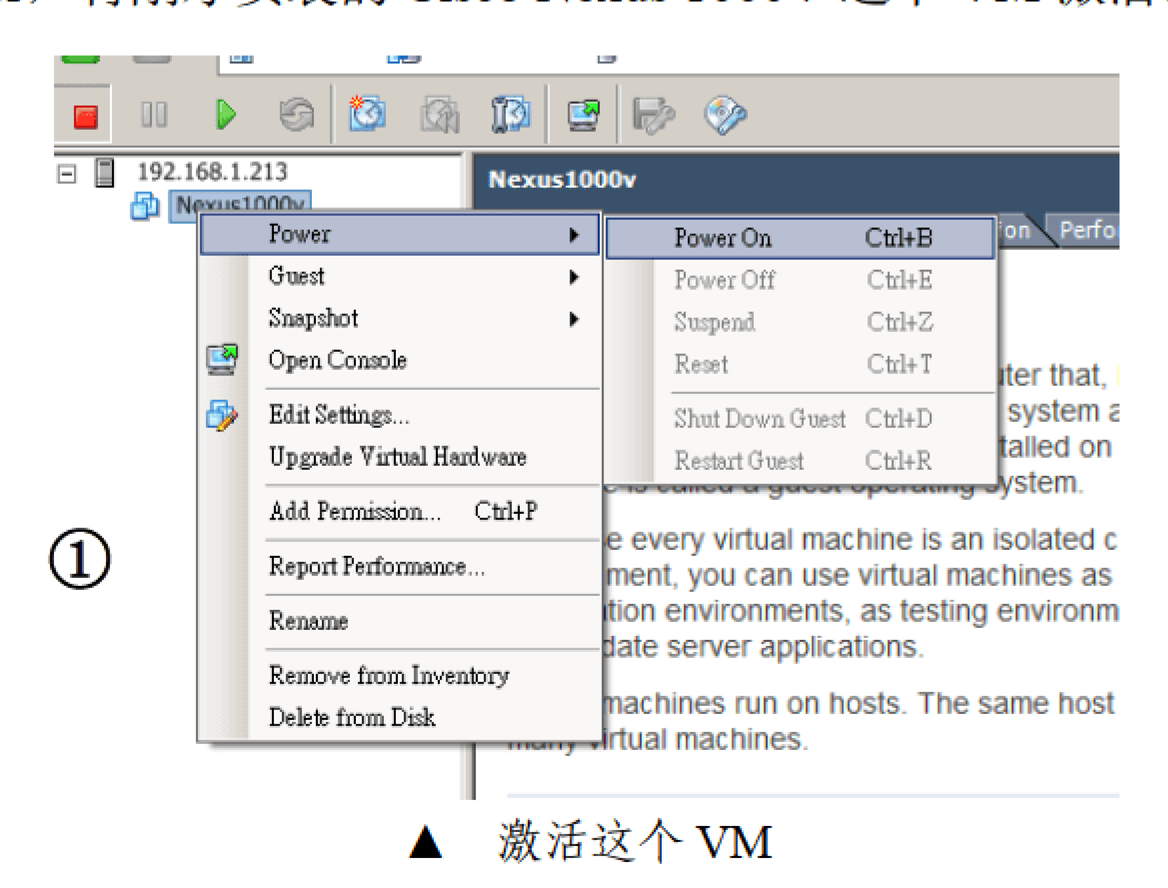
3. 配置 VSM 的管理参数

当我们安装好 VSM 的 VM 之后，接下来就是要开始使用，并且进行配置。配置 Cisco Nexus

1000V 的 VSM 可以通过网页模式或是直接进入 NX-OS 配置，笔者是强烈建议使用网页模式，文本配置常常会弹出问题，我们就看看网页配置的方式。

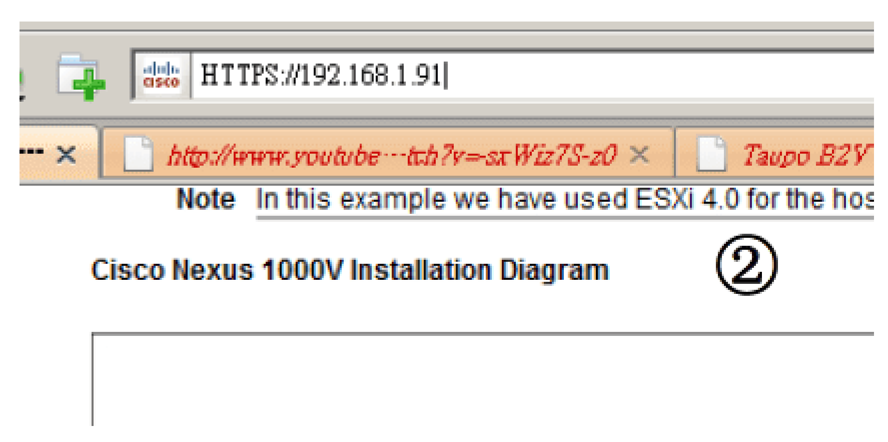
▶ 使用网页模式配置 Cisco Nexus 1000V VSM

1. 进入 vCenter，将刚才安装的 Cisco Nexus 1000V 这个 VM 激活。



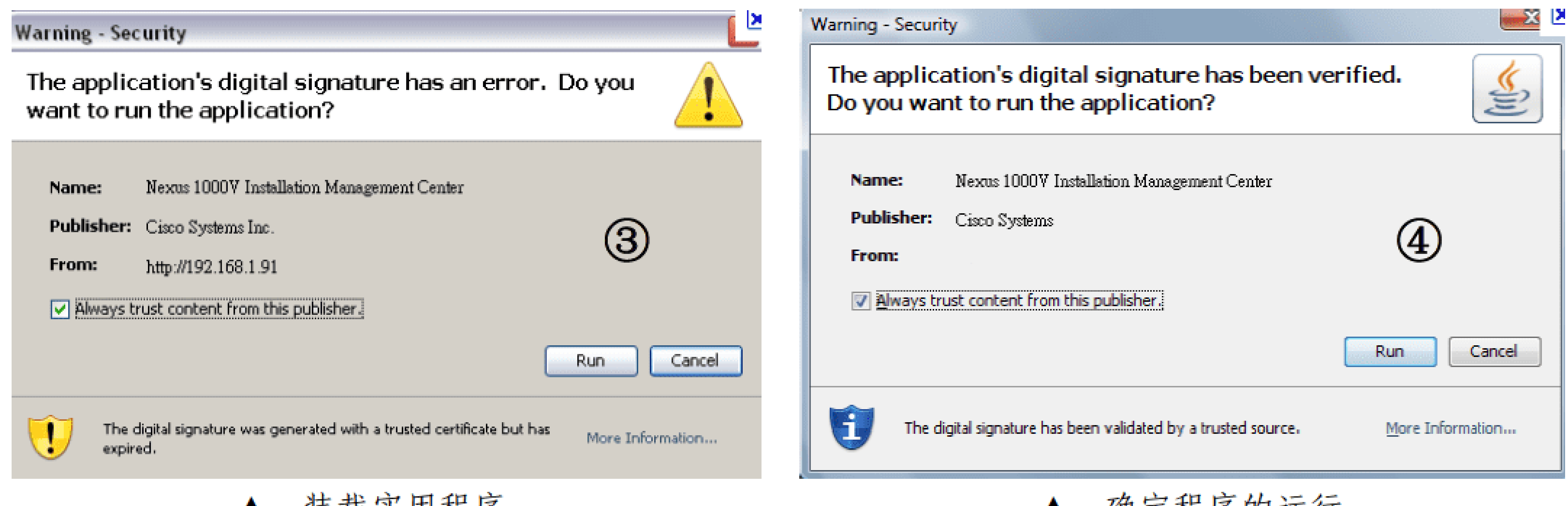
▲ 激活这个 VM

2. 确定你的浏览器支持 Java 的运行。
3. 首先我们进入浏览器，并且键入刚才配置的管理 IP。如 https://192.168.1.91。



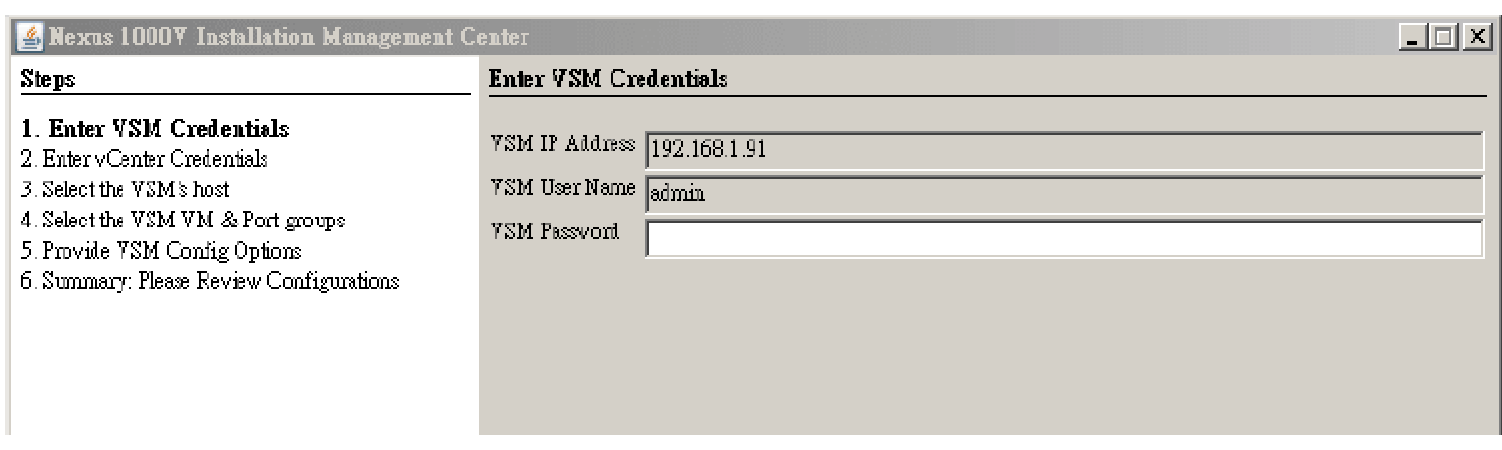
▲ 键入 VSM 的管理 IP

4. 此时会进入 Java 激活。此时 Java 虚拟机会装载 VSM 中的 Java 程序。
5. 接下来系统会要求你确定并运行这个 Java 程序。



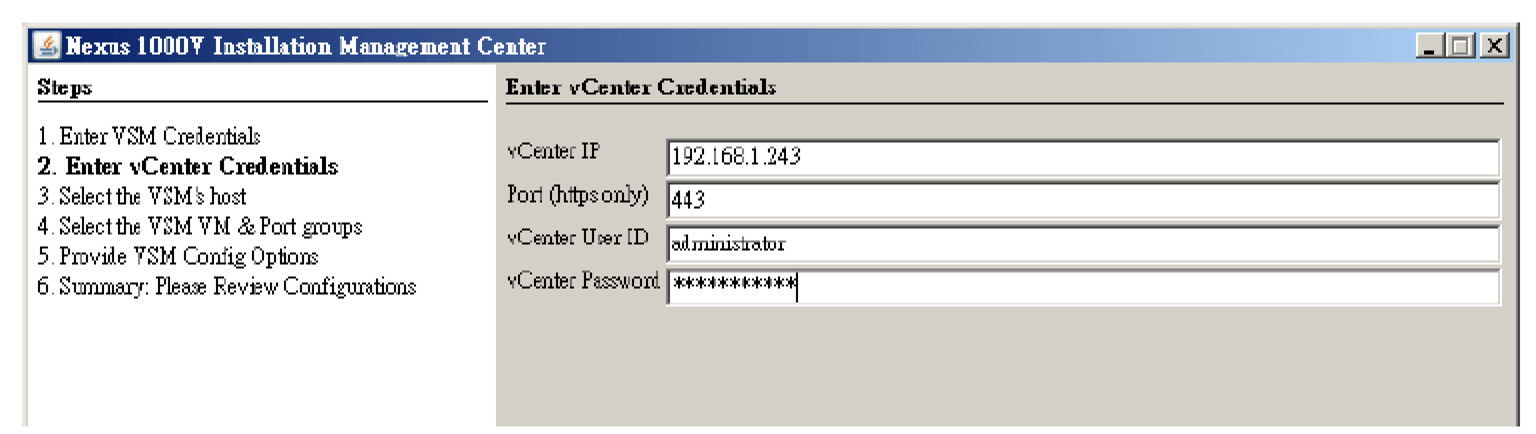
▲ 装载实用程序　　▲ 确定程序的运行

6. 激活完毕之后就会进入，并且要求你键入刚才配置的账号口令。单击 Next 按钮继续。



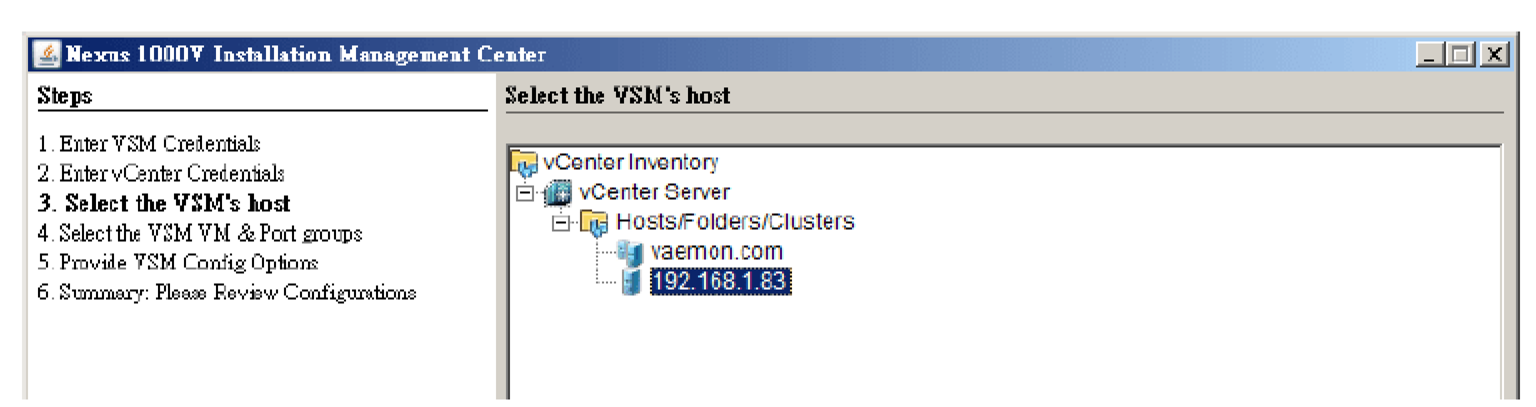
▲ 键入账号口令

7. 此时会要求你键入 vCenter 的账号口令及 IP。单击 Next 按钮继续。



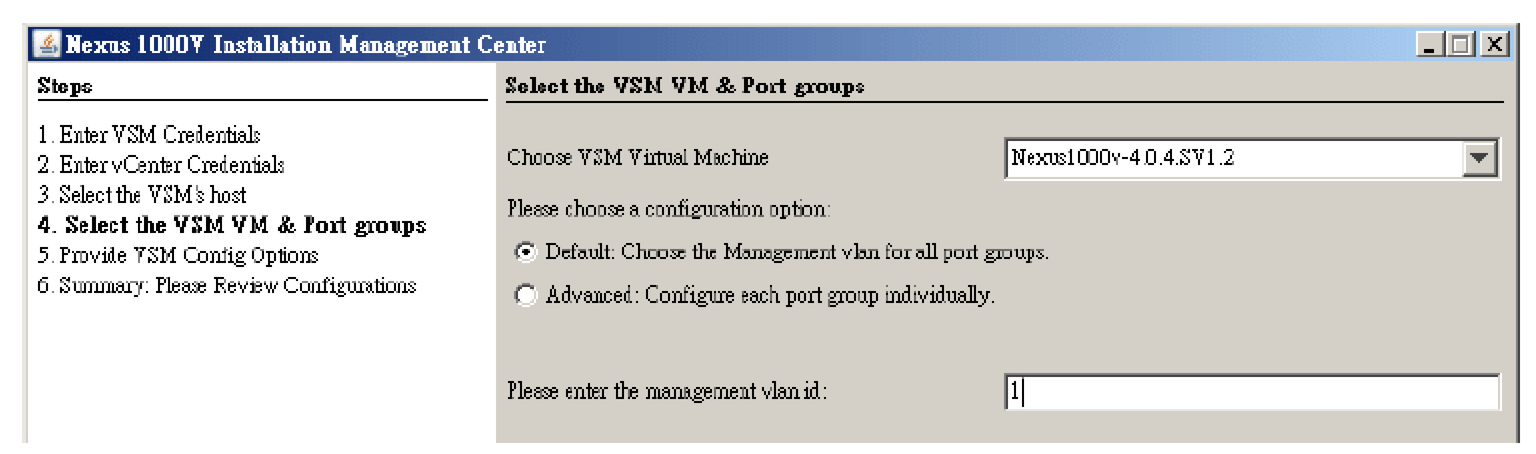
▲ 此时是键入 vCenter 的账号口令

8. 接下来选择安装 VSM 的那个 ESX 主机，单击 Next 按钮继续。



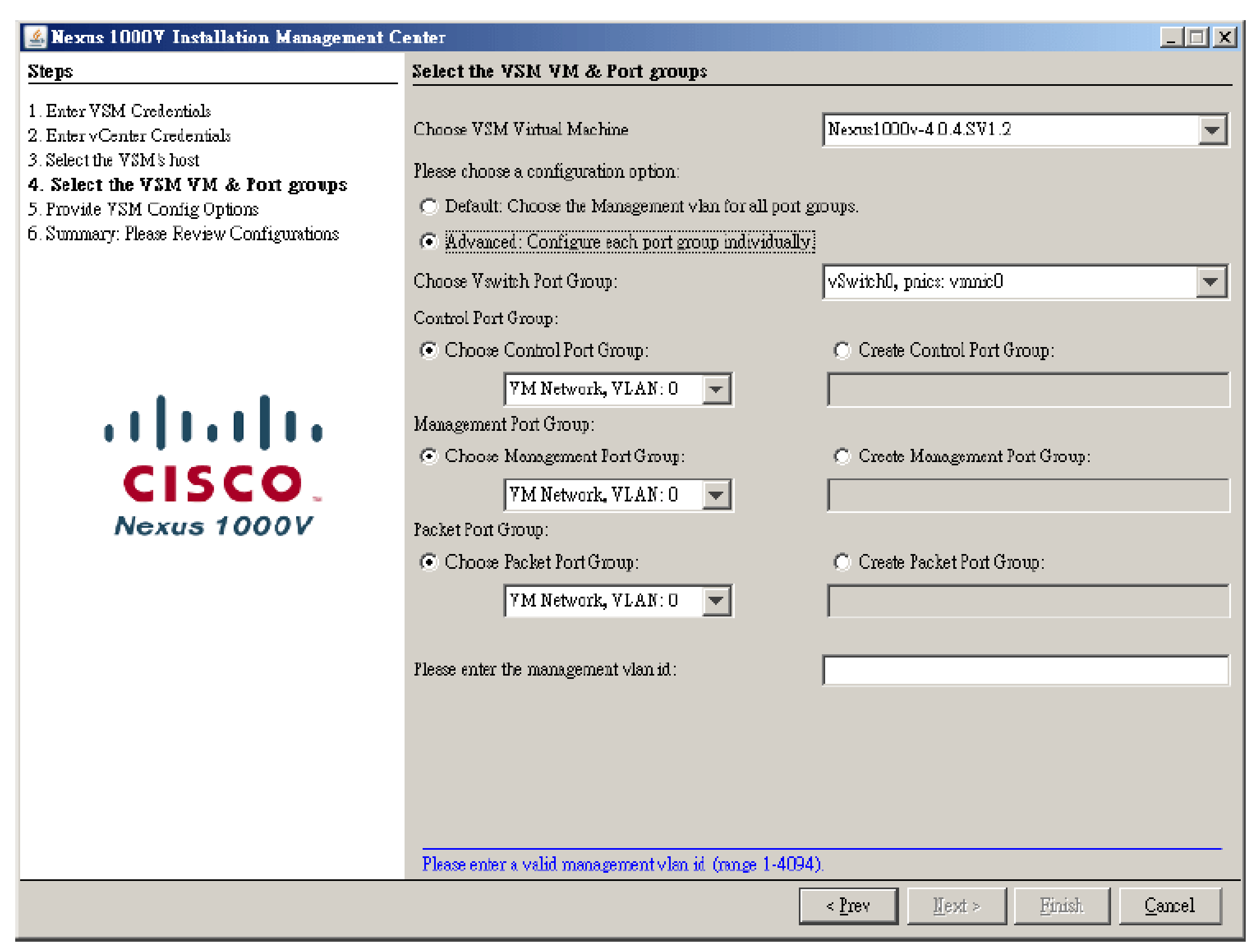
▲ 选择安装的主机

9. 接下来选择 VSM 所在的 VM，从下拉菜单中选取，并且键入配置选项。我们只要键入默认值即可。另外也要键入管理的 VLAN ID，键入完之后单击 Next 按钮继续。



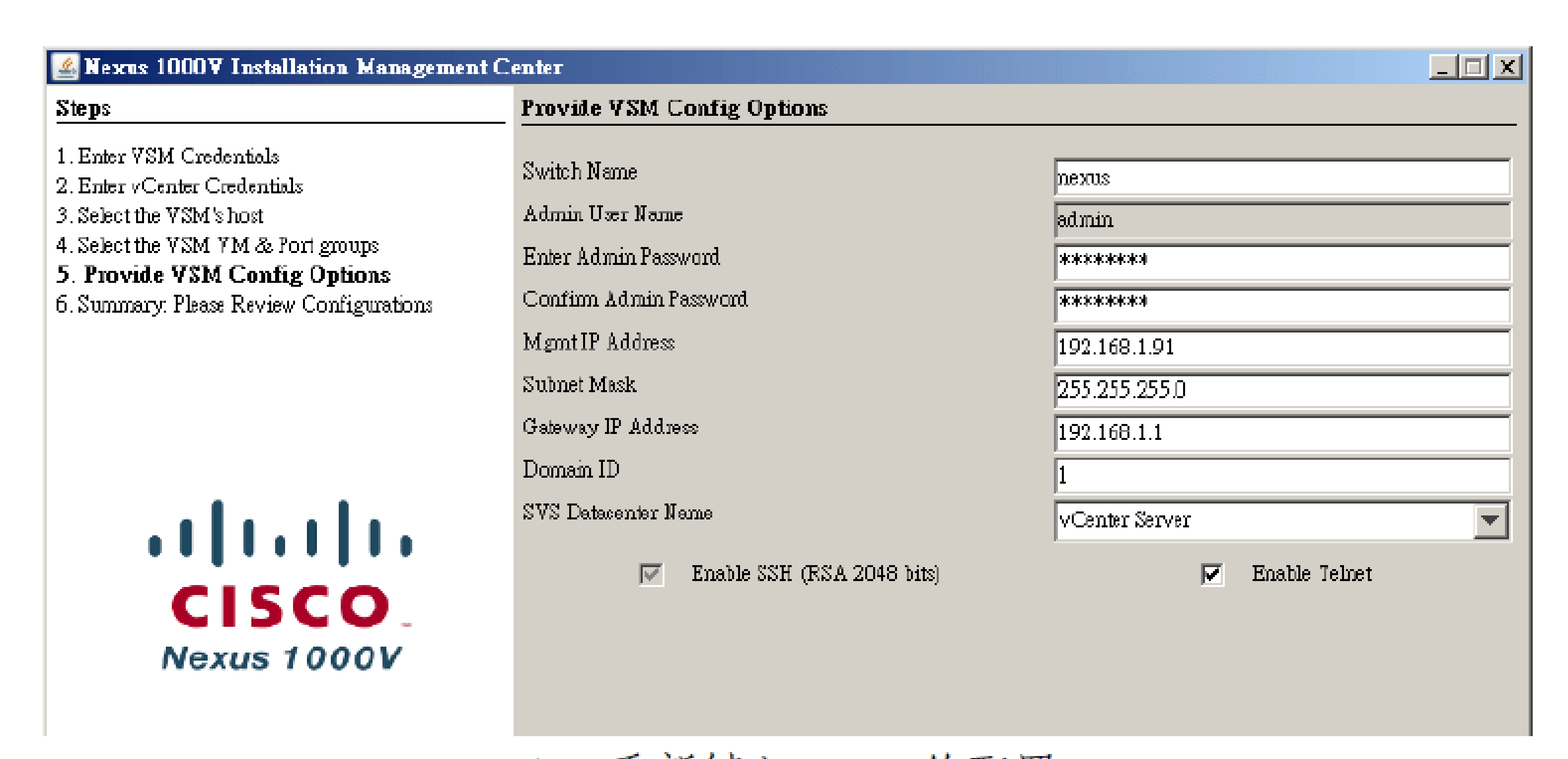
▲ 选择 VSM 和 VLAN ID

10. 如果你选择了 Advanced 配置，就会弹出 VLAN 和网络 Port Group 对应的选择，由于我们已经配置完毕，因此不需要重新对应。



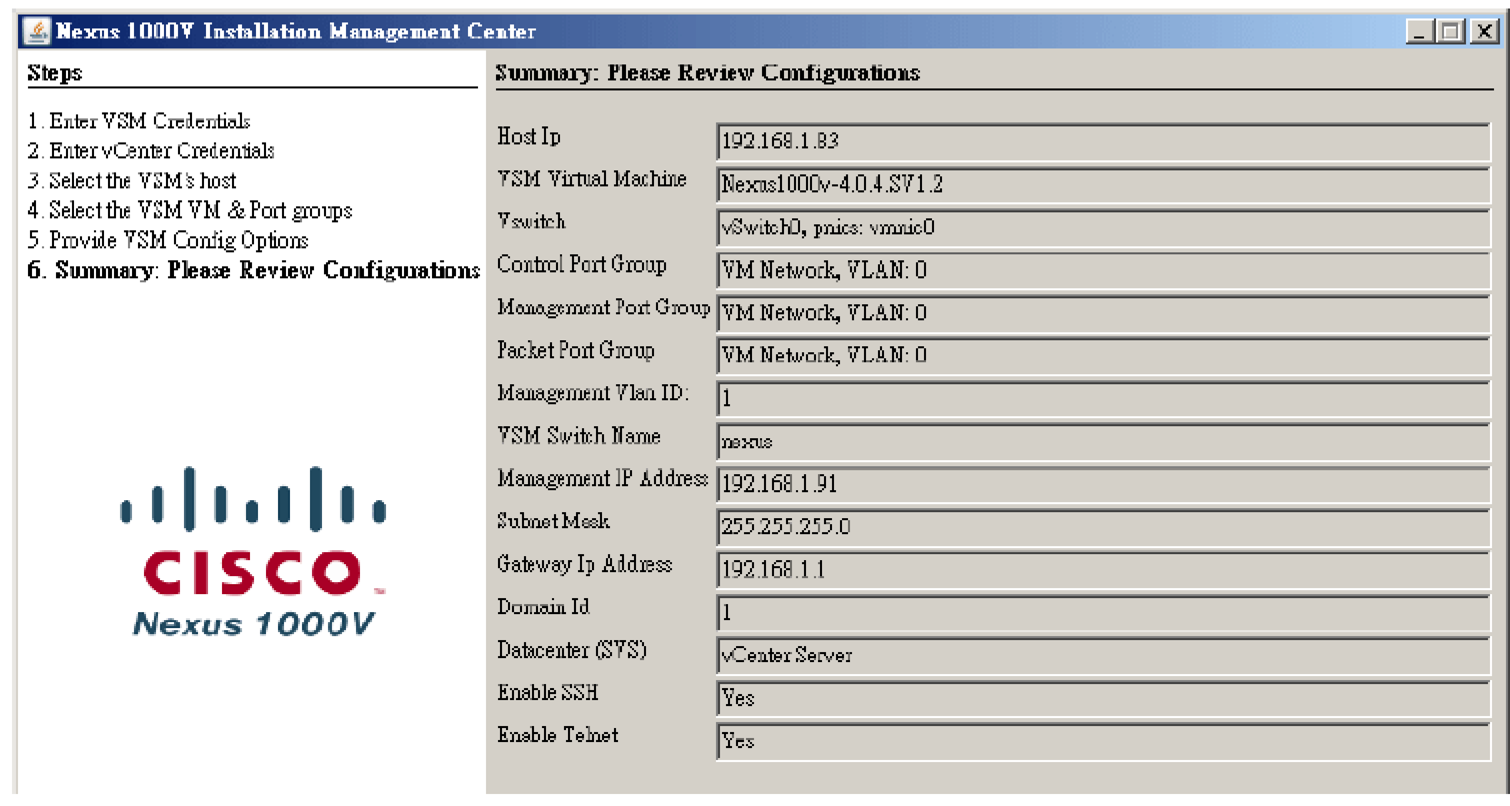
▲ 也可以进入进阶选择对应 VLAN

11. 接下来就是要配置这个在 vCenter VDS 中的交换机名称以及账号口令等数据。别忘了激活 Telnet 方便我们进入操作。键入完之后单击 Next 按钮继续。



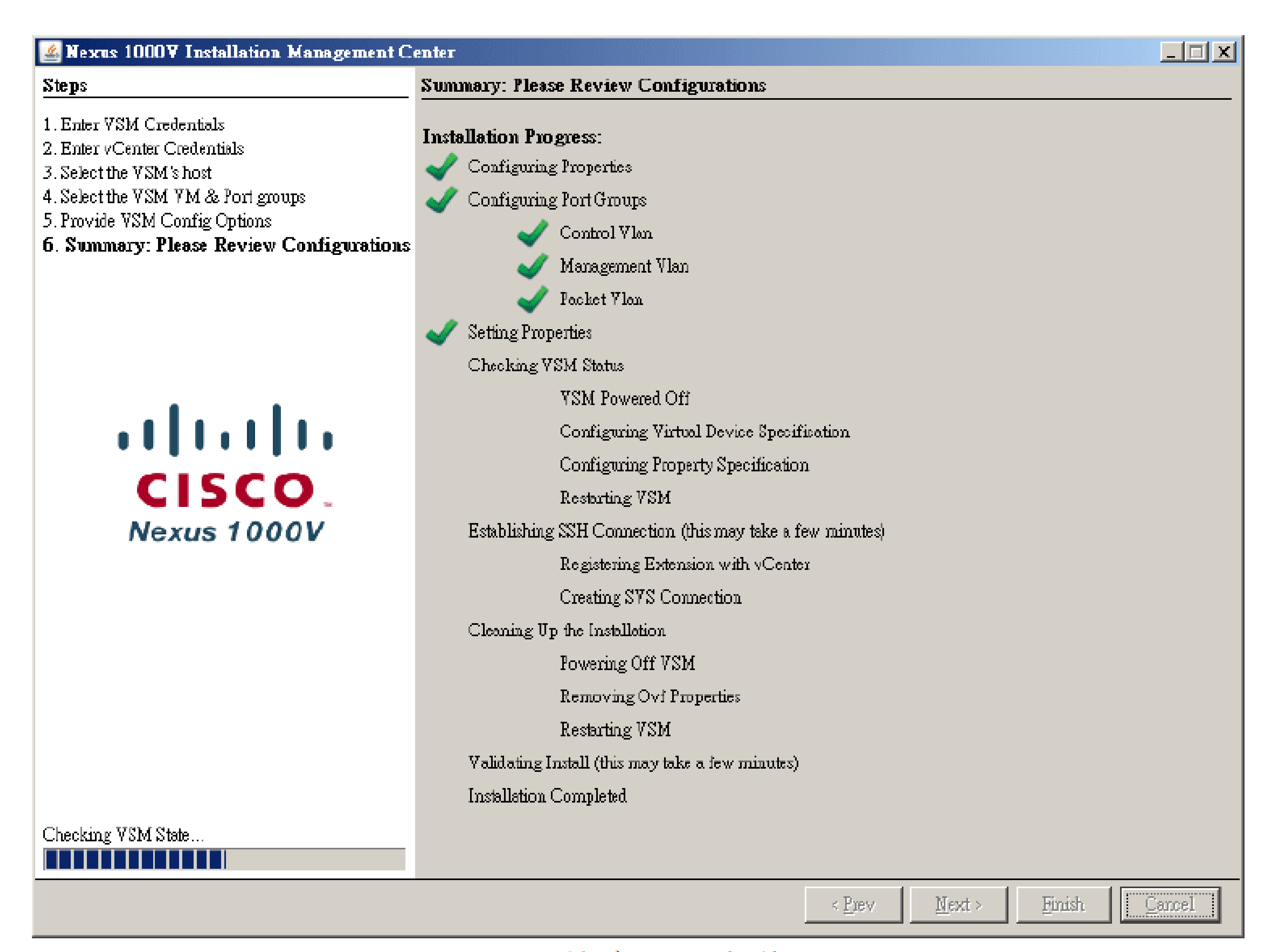
▲ 重新键入 VSM 的配置

12. 此时会进入总结画面，意味着我们的配置已经落实，单击 Finish 按钮会开始安装。



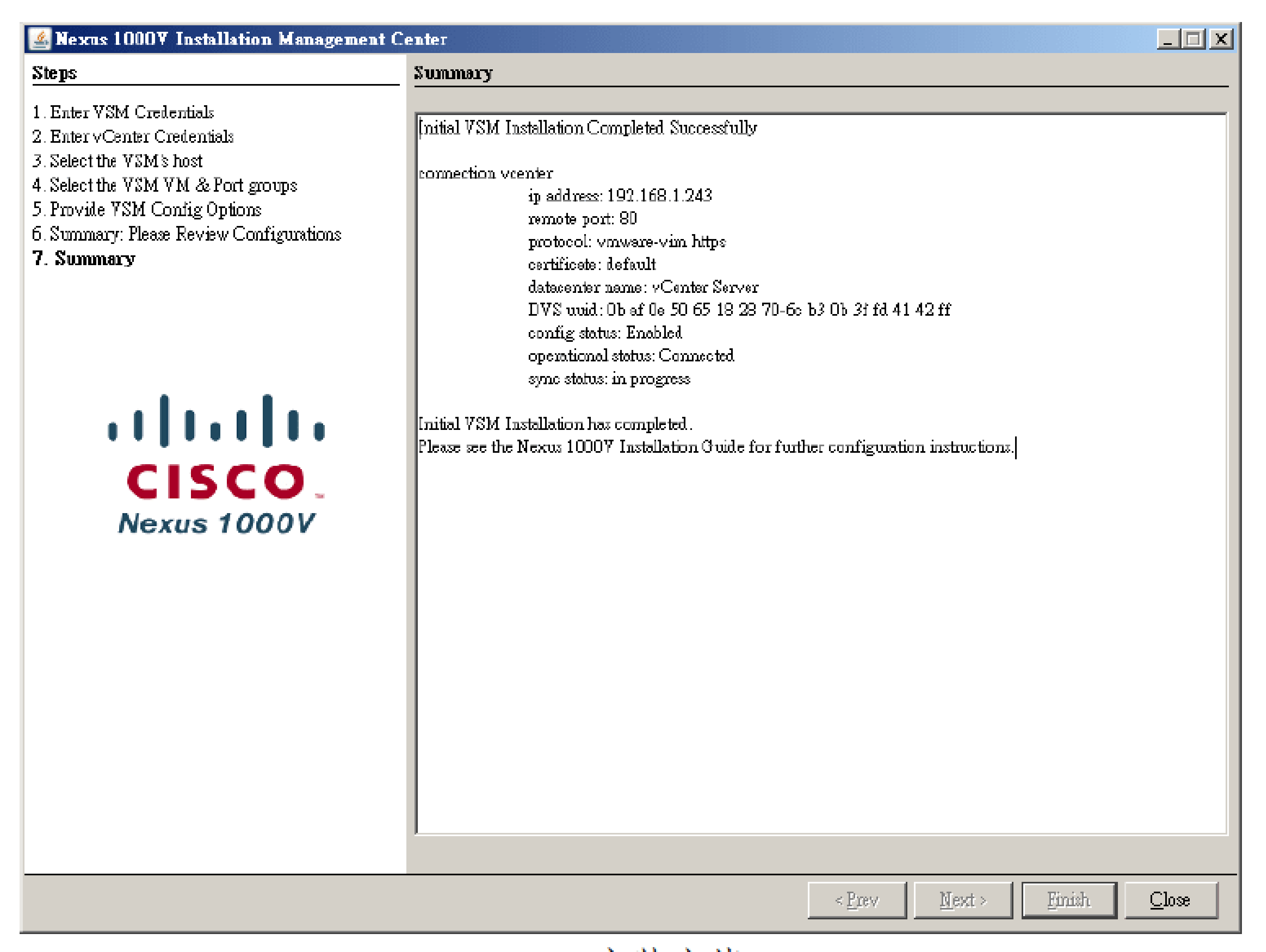
▲ 总结画面

13. 此时系统会开始遍历各个条目，并且开始安装。



▲ 检查项目安装

14. 当弹出如下图所示的画面时，意味着安装落实，你的 VSM 已经成功加入 vCenter 中的 VDS 交换机了。



▲ 安装完毕

4. 手动配置 VSM 的参数

你也可以直接进入 VSM 的 VM 中，手动配置 VSM 的参数，手动配置的情况通常是你很熟悉 Cisco 的 NX-OS，如果你是使用 Linux 环境没有 Windows 图形界面时，也只能进入用手动配置，我们就来看看。

▶ 手动配置 VSM 的参数

1. 引导进入 VSM 的 VM，并且在 vSphere Client 中打开其 Console 直接操作。

2. 通常引导需要 5～10 分钟。

3. 此时系统会要求键入 admin 的账号口令，直接键入即可。

```
— System Admin Account Setup —
Enter the password for "admin":
Confirm the password for "admin":
```

▲ 键入账号口令

4. 接下来系统会询问 HA 模式，我们需要键入“standalone”，之后按 Enter 键。

```
Enter HA role[standalone/primary/secondary]: standalone
Saving boot configuration. Please wait...
[########################################] 100%
```

▲ 键入 standalone

5. 接下来系统会询问是否进入基本配置模式，我们要键入 yes。

```
Enter HA role[standalone/primary/secondary]: standalone
Saving boot configuration. Please wait...
[########################################] 100%

— Basic System Configuration Dialog —
This setup utility will guide you through the basic configuration of the system. Setup configures only enough connectivity for management of the system.
Press Enter at anytime to skip a dialog. Use ctrl-c at anytime to skip the remaining
dialogs.

Would you like to enter the basic configuration dialog (yes/no):
```

▲ 键入 yes

6. 接下来是询问这台机器的 HA 角色，我们键入 primary。

```
Enter HA role[standalone/primary/secondary]: primary

[########################################] 100%


        ---- Basic System Configuration Dialog ----

This setup utility will guide you through the basic configuration of
the system. Setup configures only enough connectivity for management
of the system.

*Note: setup is mainly used for configuring the system initially,
when no configuration is present. So setup always assumes system
defaults and not the current system configuration values.

Press Enter at anytime to skip a dialog. Use ctrl-c at anytime
to skip the remaining dialogs.

Would you like to enter the basic configuration dialog (yes/no):
```

▲ 键入 primary

7. 接下来系统会询问是否再进入基本配置交谈，键入 yes。

```
Press Enter at anytime to skip a dialog. Use ctrl-c at anytime
to skip the remaining dialogs.

Would you like to enter the basic configuration dialog (yes/no):
```

▲ 键入 yes

8. 接下来系统询问是否要创建第二个账号口令，键入 no。

```
Create another login account (yes/no) [n]: no
```

▲ 不要创建第二个账号

9. 接下来询问是否要创建只读的 SNMP，键入 no。

```
Configure read-only SNMP community string (yes/no) [n]: no
```

▲ 键入 no

10. 接下来是只读的 SNMP 字符串，键入 no。

```
Configure read-write SNMP community string (yes/no) [n]: no
```

▲ 也是键入 no

11. 接下来是替这个 vCenter 中的交换机取一个名字，如 n1000v。

```
Enter the switch name: n1000v
```

▲ 替交换机取名字

12. 接下来是配置管理 IP，键入 yes 之后再键入 IP。

```
Continue with Out-of-band (mgmt0) management configuration? [yes/no]: yes
Mgmt0 IPv4 address: 172.28.15.152
Mgmt0 IPv4 netmask: 255.255.255.0
```

▲ 键入管理 IP

13. 接下来是键入网关 IP。

```
Configure the default-gateway: (yes/no) [y]: yes
IPv4 address of the default-gateway: 172.28.15.1
```

▲ 网关 IP

14. 接下来是选择是否键入进阶 IP 选择，键入 no。

```
Configure Advanced IP options (yes/no)? [n]: no
```

▲ 不键入进阶选项

15. 接下来是选择是否激活 Telnet，键入 yes。

```
Enable the telnet service? (yes/no) [y]: yes
```

▲ 激活 Telnet

16. 接下来是 SSH 的激活，我们选择 yes 并且照着图中键入。

```
Enable the ssh service? (yes/no) [y]: yes
Type of ssh key you would like to generate (dsa/rsa) : rsa
Number of  key bits <768-2048> : 1024
```

▲ 激活 SSH

17. 接下来是 NTP 的配置，不用配置。

```
Configure NTP server? (yes/no) [n]: no
```

▲ 不配置 NTP

18. 接下来是报文和总控的 VLAN 配置，我们就照着图中键入。

```
Configure n1000v domain parameters: yes
Enter control vlan <1-3967, 4048-4093>: 260
Enter packet vlan  <1-3967, 4048-4093>: 261
Enter domain id <1-4095>: [152]:
```

▲ 配置 VLAN

19. 接下来系统会进入总结，系统会询问是否要重新键入，我们就选择不要。

```
Would you like to edit the configuration? (yes/no) [n]:no
```

▲ 不要重新键入

20. 最后系统会询问是否要存盘，键入 yes 之后即配置落实。

```
Use this configuration and save it? (yes/no) [y]: yes
[########################################] 100%
```

▲ 将配置存盘

5. 取得 VSM 的用户许可证

要常规使用 VSM 必须取得授权，Cisco 提供了试用版授权使用 60 天，并且提供全功能，我们必须先取得 VSM 的主机号，再到 Cisco 的网站下载。当然你必须有 Cisco 网站的注册身份才行。我们就来看看完整的取得步骤。

▶ 取得 VSM 的评估授权

1. 进入 VSM VM 的主控台中，并且键入 show license host-id。此时会弹出 ID。

```
n1000v# show license host-id
License hostid: VDH=1280389551234985805
```

▲ 获取 host-id

2. 进入 Cisco 的授权页面，遵照网站的步骤键入该 ID，并且取得许可证文件。许可证文件会以邮箱的方式寄回。

3. 获取文件之后，放在 VSM 能取得的地方，并且将这个文件克隆到 VSM 中，可以使用 SCP。

```
n1000v@ copy scp://user@linux-box.cisco.com/home/user/n1kv_license.lic bootflash:
Enter vrf (If no input, current vrf 'default' is considered):
user@linux-box.cisco.com's password:
n1kv_license.lic                                  100%  252     0.3KB/s   00:00

n1000v@
```

▲ 克隆文件

6. 安装许可证文件到 VSM 中

在取得授权之后，接下来就是安装到 VSM 中，我们就来看看。

▶ 将许可证文件安装

1. 进入 VSM 的 VM 中。
2. 键入 install license bootflash:license_file.lic。

```
n1000v# install license bootflash:license_file.lic
Installing license ..done
n1000v#
```

▲ 安装文件

3. 键入 show license file license.lic 来检视许可证文件。

```
n1000v# show license file license.lic
SERVER this_host ANY
VENDOR cisco
INCREMENT NEXUS1000V_LAN_SERVICES_PKG
cisco 1.0 permanent 1 \
        HOSTID=VDH=1575337335122974806 \

NOTICE="<LicFileID>license.lic</LicFileI
D><LicLineID>0</LicLineID> \
        <PAK>dummyPak</PAK>"
SIGN=3AF5C2D26E1A
n1000v#
```

▲ 检视授权档

4. 键入 show license usage NEXUS1000V_LAN_SERVICES_PKG。

```
Example:
n1000v# show license usage
NEXUS1000V_LAN_SERVICES_PKG
Application
-----------
VEM 3 - Socket 1
VEM 3 - Socket 2
VEM 4 - Socket 1
VEM 4 - Socket 2
-----------


n1000v#
```

▲ 检视 PKG 文件

5. 将运行中的配置文件克隆成激活档，键入 copy running-config startup-config，键入完毕即告落实。

```
n1000v(config)# copy running-config startup-config
```

▲ 激活运行

6. 当我们键入完毕之后，VSM 应该可以连接至 vCenter 了。我们可以使用 ping 来测试看是否能连接到外网，能连接成功即告落实。

```
n1000v# ping 172.28.15.1

PING 172.28.15.1 (172.28.15.1): 56 data bytes

Request 0 timed out

64 bytes from 172.28.15.1: icmp_seq=1 ttl=63 time=0.739 ms
64 bytes from 172.28.15.1: icmp_seq=2 ttl=63 time=0.197 ms
64 bytes from 172.28.15.1: icmp_seq=3 ttl=63 time=0.224 ms
64 bytes from 172.28.15.1: icmp_seq=4 ttl=63 time=0.67 ms

--- 172.28.15.1 ping statistics ---

5 packets transmitted, 4 packets received, 20.00% packet loss

round-trip min/avg/max = 0.517/0.114/0.711 ms
```

▲ 可成功连至外网

23.2.3 配置 vCenter 上的 Cisco Nexus 1000V 部件

安装好 VSM 之后，接下来的任务就是在 vCenter 中配置这个 Cisco Nexus 1000V 的 Plugin 方便操作。在 Plugin 配置好之后，就可以安装 VEM 了。

1. 安装 vCenter Plugin

为了方便操作 Cisco Nexus 1000V，我们可以在 vCenter 中安装 Plugin，下面就是完整的步骤。

▶ 在 vCenter 中安装 Cisco Nexus 1000V 的 Plugin

1. 从浏览器进入 VSM 的 IP，不需要 https，使用 http 即可。
2. 从中下载 cisco_nexus_1000v_extension.xml。

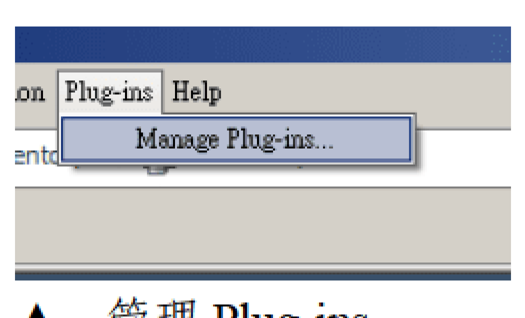


Index of /

mode	links	bytes	last-changed	name
dr-x	3	120	Jan 17 09:13	./
drwx	12	240	Jan 17 11:41	../
-r--	1	46	Jan 17 09:13	bundle_id.txt
dr-x	3	120	Jan 17 09:13	cisco/
-rw-	1	656	Jan 17 11:42	cisco_nexus_1000v_extension.xm
-r--	1	228	Jan 17 11:49	index.xml

▲ 下载 XML 文件

3. 从 Plug-in 选项卡中选择 Manage Plug-ins 选项。
4. 在空白处右击选择 New Plug-in 选项。

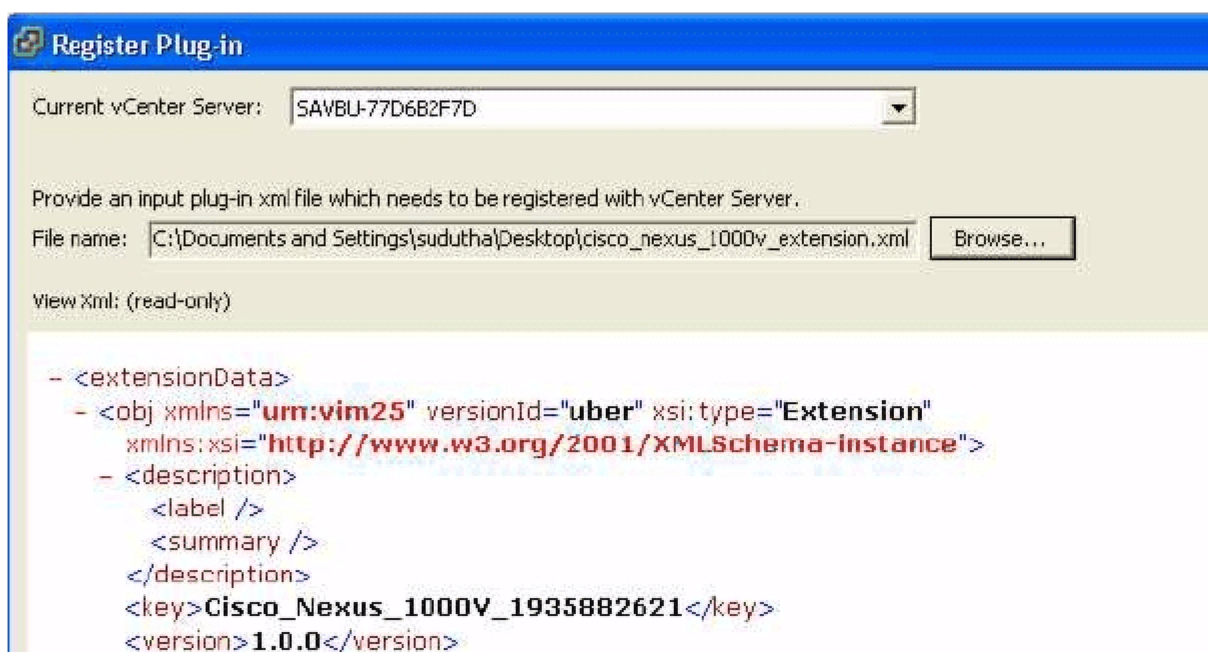


▲ 管理 Plug-ins



▲ 新建

5. 装载刚才下载的文件。



▲ 装载文件

6. 注册新的 Plug-in。



▲ 要注册后才能使用

7. 弹出安全报警窗口时，单击 Ignore 按钮。



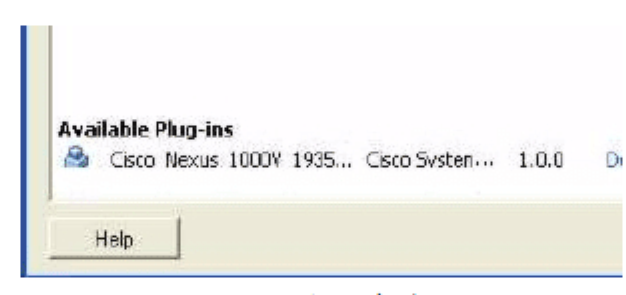
▲ 略过安全报警

8. 当弹出下图中的画面时，单击 OK 按钮即落实。



▲ 安装落实

9. 你可以看到图中的 Plug-in 已经安装落实。



▲ 已经弹出了

2. 联机至 vCenter Server

接下来我们就要将 VSM 联机至 vCenter 中，才能继续安装 VEM 和给 VM 使用，下面就是详细的步骤。

▶ **联机至 vCenter**

1. 进入 VSM 这个 VM 的主控台。
2. 键入下列的命令：

▶ **联机至 vCenter 的命令**

```
n1000v# config t
n1000v (config) # svs connection VC
n1000v ( (config-svs-conn) # vmware dvs datacenter-name Vaemon.com
n1000v ( (config-svs-conn) # protocol vmware-vim
n1000v ( (config-svs-conn) # remote ip address 172.28.15.150
```

```
n1000v# config t
n1000v(config)# svs connection VC
n1000v((config-svs-conn)# vmware dvs datacenter-name Vaemon.com
n1000v((config-svs-conn)# protocol vmware-vim
n1000v((config-svs-conn)# remote ip address 172.28.15.150
```

▲ 键入命令

3. 键入下面的命令联机到 vCenter。

▶ **联机到 vCenter**

```
n1000v# config t
n1000v (config) # svs connection VC
n1000v (config-svs-conn) # connect
n1000v (config-svs-conn) #
```

```
n1000v# config t
n1000v(config)# svs connection VC
n1000v(config-svs-conn)# connect
n1000v(config-svs-conn)#
```

▲ 联机到 vCenter

4. 键入完之后可以确定连接，键入 show svs connections。

```
n1000v# show svs connections


connection VC:
   hostname: 172.28.15.150
   protocol: vmware-vim https
   certificate: default
   datacenter name: Vaemon.com
   DVS uuid: 5f 1d 27 52 34 42 01 63-b1 a2 20 4a 36 3b ac f4
   config status: Enabled

   operational status: Connected
```

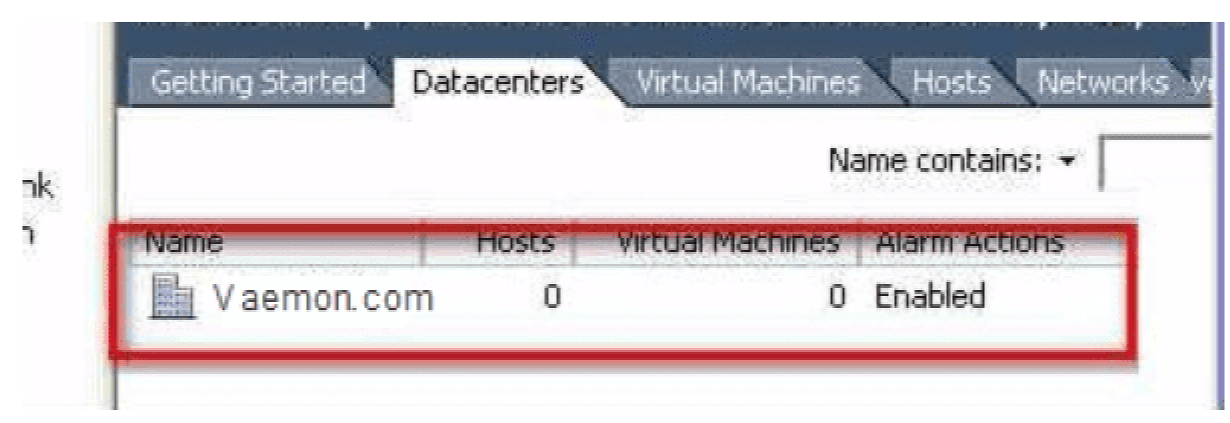
▲ 显示确定连接

5. 键入 copy run start 进入引导时运行选项。

```
n1000v(config-svs-conn)# copy run start
[########################################] 100%
n1000v(config-svs-conn)#
```

▲ 引导时运行

6. 我们在 vCenter 中就可以看到 vCenter 已经在该交换机中弹出了。



▲ 弹出了

23.2.4 创建 VEM 的连接

我们配置好 VSM、vCenter 中的配置后，意味着这个 VDS 已经和 vCenter 之间创建交换机关联了。接下来我们要到每一个 ESX 中分别配置 VEM 的连接。但是在操作之前，我们得先作为 VLAN，并且将 VEM 的配置文件准备好，如此一来，就可以在 ESX 上线后，直接克隆配置文件而连接至 VSM 了。

1. 创建 VLAN

Cisco Nexus 1000V 创建在不同的 VLAN 上进行总控、报文和管理的传输，我们在 VSM 中已经创建了 VLAN 的连接，因此这里要进行物理交换机 VLAN 的配置。Cisco Nexus 1000V 支持 4094 个 VLAN，其中编号 1 是 Cisco Nexus 1000V 自身使用的，而 2-1005 是可以自由使用。从 1006 至 4094 则是扩展 VLAN，虽然可以自由使用，但无法关闭，而编号也自动生成。我们前面已经定义了编号，就直接使用。

▶ 创建 VLAN

（1）键入 VSM 中。

（2）键入 config t。

（3）键入 vlan 5，创建编号为 5 的 VLAN。

（4）键入 vlan 15-20，创建编号为 15-20 的 VLAN。

（5）键入 copy running-config startup-config 将 VLAN 选项放入引导文件中。

2. 安装 VEM

我们可以在单台的 ESX 中安装 VEM，使用的是 VMware 的 Update Manager。在安装前，确定你的 VUM 已经可以正确使用了。下面就是步骤。

▶ 在 ESX 中安装 VEM

（1）先进入浏览器，进入管理 IP，并且下载 VEM 的文件。

（2）使用 SSH 程序进入 ESX 主机中，如 PuTTY。

（3）键入下列程序进行安装：

▶ 安装 VEM

```
esxcfg-firewall -allowoutgoing
```

```
esxupdate --bundle http://<<管理 IP>/cisco/cisco-vem-v100-4.0.4.1.1.27-0.4.2.
          zip info
esxupdate --bundle  http://<管理 IP>/cisco/cisco-vem-v100-4.0.4.1.1.27-0.4.2
          .zip update
esxcfg-firewall -blockoutgoing
esxupdate query
```

键入下面的程序遍历是否有配置好 DPA：

▶ 遍历 DPA 的状态

```
/etc/init.d/vem-v100-vssnet-dpa status
/etc/init.d/vem-v100-vssnet-load status
vem status
```

3. 创建 VSM-VEM 连接的系统配置文件

当我们创建好 VLAN 之后，接下来就是要让每一个运行 VEM 的 ESX 能和 VSM 这台主机连接。由于多台 ESX 一台台配置十分麻烦，因此我们选择创建一个系统配置文件来给多台 ESX 上的 VEM 使用，当然是使用 VLAN，下面就是步骤。

▶ 创建 VSM-VEM 连接的配置文件

（1）还是进入 VSM 的 VM 之中。

（2）键入下列的命令：

▶ 创建 VSM-VEM 连接命令

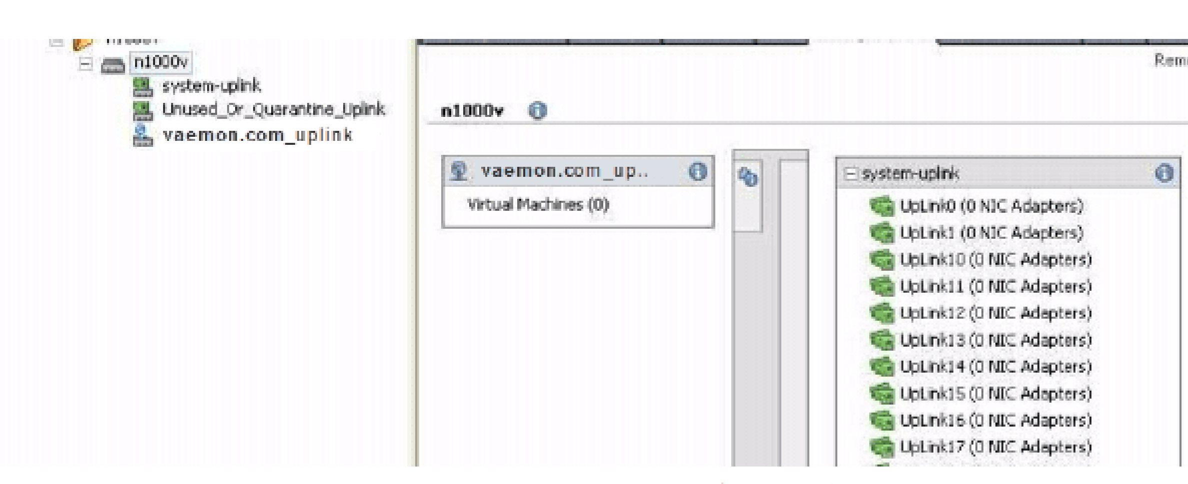
```
n1000v# config t
n1000v (config) # port-profile system-uplink
n1000v (config-port-prof) # switchport mode trunk
n1000v (config-port-prof) # switchport trunk allowed vlan 260-261
n1000v (config-port-prof) # no shut
n1000v (config-port-prof) # system vlan 260, 261
n1000v (config-port-prof) # vmware port-group
n1000v (config-port-prof) # capability uplink
n1000v (config-port-prof) # state enabled
n1000v (config-port-prof) # copy run start
```

```
n1000v# config t
n1000v(config)# port-profile system-uplink
n1000v(config-port-prof)# switchport mode trunk
n1000v(config-port-prof)# switchport trunk allowed vlan 260-261
n1000v(config-port-prof)# no shut
n1000v(config-port-prof)# system vlan 260, 261
n1000v(config-port-prof)# vmware port-group
n1000v(config-port-prof)# capability uplink
n1000v(config-port-prof)# state enabled
n1000v(config-port-prof)# copy run start

[########################################] 100%

n1000v(config-port-prof)#
```

▲ 创建连接



▲ 创建落实

4. 创建 VM 流量的 Uplink 通信端口配置文件

接下来我们要创建 VM 流量的配置文件。

▶ 创建 VM 流量的配置文件

```
n1000v$ config t
n1000v (config) # port-profile vm-uplink
n1000v (config-port-prof) # switchport mode access
n1000v (config-port-prof) # capability uplink
n1000v (config-port-prof) # switchport access vlan 262
n1000v (config-port-prof) # vmware port-group
n1000v (config-port-prof) # no shut
n1000v (config-port-prof) # state enabled
n1000v (config-port-prof) # copy run start
```

```
n1000v$ config t
n1000v(config)# port-profile vm-uplink
n1000v(config-port-prof)# switchport mode access
n1000v(config-port-prof)# capability uplink
n1000v(config-port-prof)# switchport access vlan 262
n1000v(config-port-prof)# vmware port-group
n1000v(config-port-prof)# no shut
n1000v(config-port-prof)# state enabled
n1000v(config-port-prof)# copy run start

[########################################] 100%

n1000v(config-port-prof)#
```

▲ 创建 up_link

5. 创建 VM 流量的数据通信端口配置文件

接下来最后一步，就是创建数据通信端口的配置文件。

▶ 创建 VM 流量的数据通信端口配置文件

```
n1000v$ config t
n1000v (config) # port-profile data262
n1000v (config-port-prof) # switchport mode access
n1000v (config-port-prof) # switchport access vlan 262
n1000v (config-port-prof) # vmware port-group data262
n1000v (config-port-prof) # no shut
n1000v (config-port-prof) # state enabled
n1000v (config-port-prof) # copy run start
```

```
n1000v$ config t
n1000v(config)# port-profile data262
n1000v(config-port-prof)# switchport mode access
n1000v(config-port-prof)# switchport access vlan 262
n1000v(config-port-prof)# vmware port-group data262
n1000v(config-port-prof)# no shut
n1000v(config-port-prof)# state enabled
n1000v(config-port-prof)# copy run start

[########################################] 100%

n1000v(config-port-prof)#
```

▲ 创建 DATA 连接

6. 将 ESX 主机加入 VDS 中

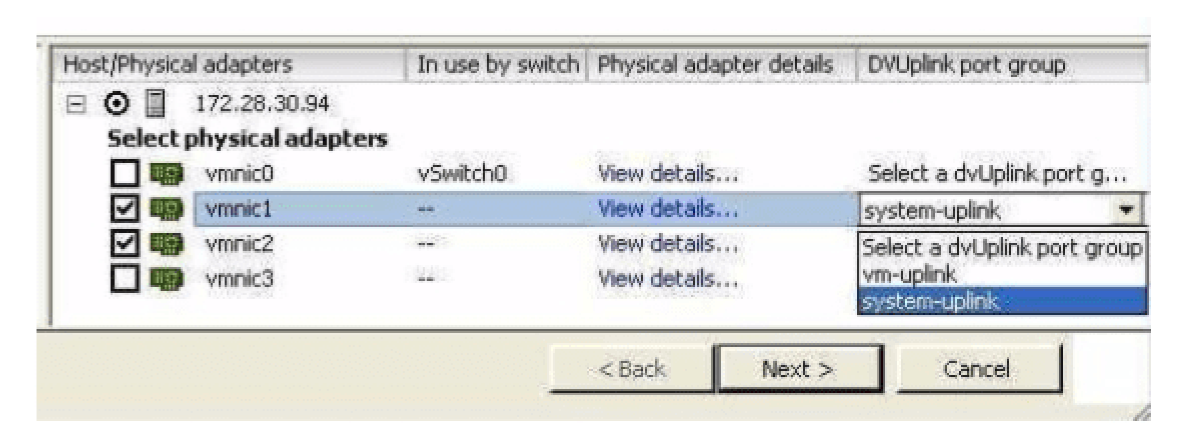
接下来最重要的事，就是将 ESX 主机加入刚才创建的 VSM 中了。我们可以从上面的关系图看出来，VSM 通过和 VEM 的 Uplink 创建连接并且接手管理。当你的 ESX 主机被 Cisco Nexus 1000V 的 VSM 管理之后，接下来加入 VM 或是连接上网络的功能也就更多了。我们首先将刚才在 Cisco Nexus 1000V 中创建的 VMNIC 对应好，分别如表 2 所示。

VMNIC 和流量 VLAN 的对应

```
VLAN
VMNIC
Control VLAN
System-uplink VMNIC
Packet VLAN
System-uplink VMNIC
VM Data
VM-uplink Port Group
```

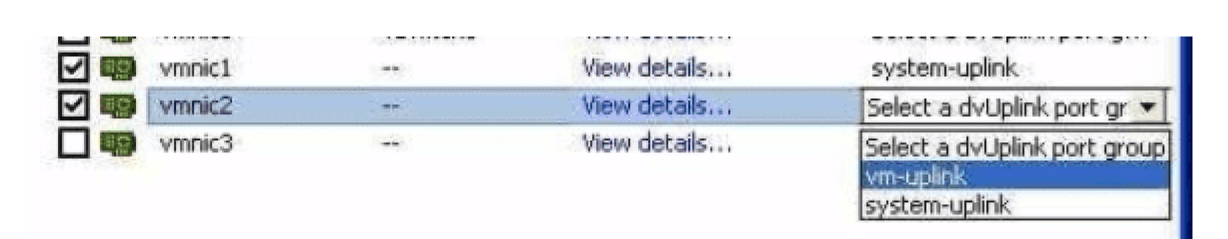
▶ 将 ESX 主机加入 Cisco Nexus 1000V 中

1. 进入 vSphere Client 中，并且选择 Inventory Networking。
2. 在刚才创建的交换机中右击，选择 Add Host 选项。
3. 此时会要求你选择主机和该主机对应的 Cisco Nexus 1000V 网络 VMNIC。我们将 VMNIC 对应到正确的 Uplink 上。键入完之后单击 Next 按钮继续。



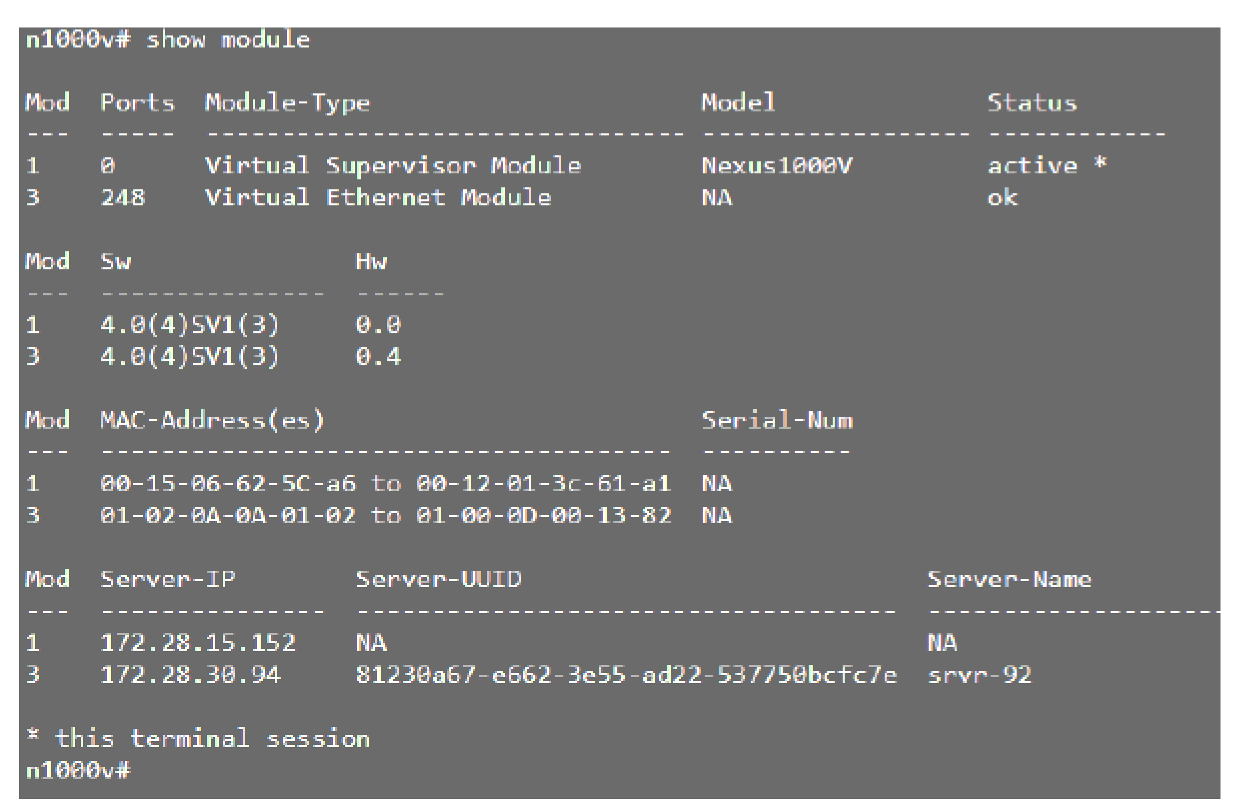
▲ 创建网卡的连接

4. 也将第二片网卡对应到 vm-uplink 上。键入完之后单击 Next 按钮继续。



▲ 第二片网卡的连接

5. 在弹出总结画面之后，单击 Finish 按钮落实安装。
6. 当我们创建完连接之后，进入 VSM 的 VM 中，并且键入 show module 可以检视 VEM 是否和 VSM 连接了。



```
n1000v# show module

Mod  Ports  Module-Type                       Model               Status
---  -----  --------------------------------  ------------------  ------------
1    0      Virtual Supervisor Module         Nexus1000V          active *
3    248    Virtual Ethernet Module           NA                  ok

Mod  Sw               Hw
---  ---------------  ------
1    4.0(4)SV1(3)     0.0
3    4.0(4)SV1(3)     0.4

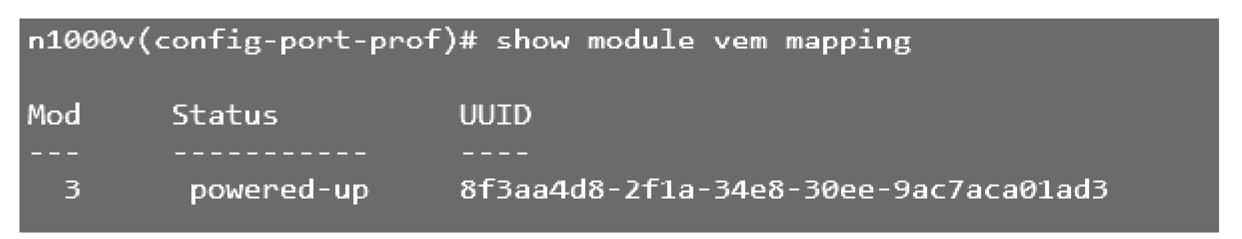
Mod  MAC-Address(es)                         Serial-Num
---  --------------------------------------  ----------
1    00-15-06-62-5C-a6 to 00-12-01-3c-61-a1  NA
3    01-02-0A-0A-01-02 to 01-00-0D-00-13-82  NA

Mod  Server-IP        Server-UUID                           Server-Name
---  ---------------  ------------------------------------  --------------------
1    172.28.15.152    NA                                    NA
3    172.28.30.94     81230a67-e662-3e55-ad22-537750bcfc7e  srvr-92

* this terminal session
n1000v#
```

▲ 确定连接

7. 接下来键入 show module vem mapping 来遍历对应的情况。

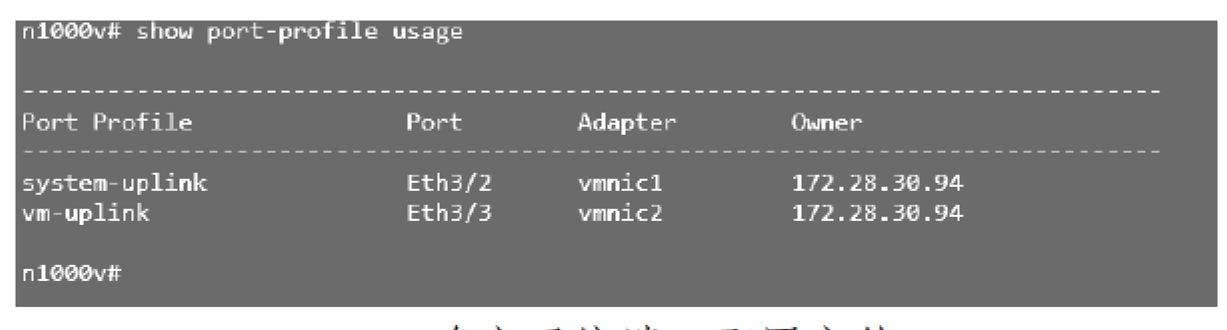


```
n1000v(config-port-prof)# show module vem mapping

Mod     Status          UUID
---     -----------     ----
  3      powered-up     8f3aa4d8-2f1a-34e8-30ee-9ac7aca01ad3
```

▲ 确定对应

8. 在 VSM 上键入 show port-profile usage 来遍历配置文件的使用状况。



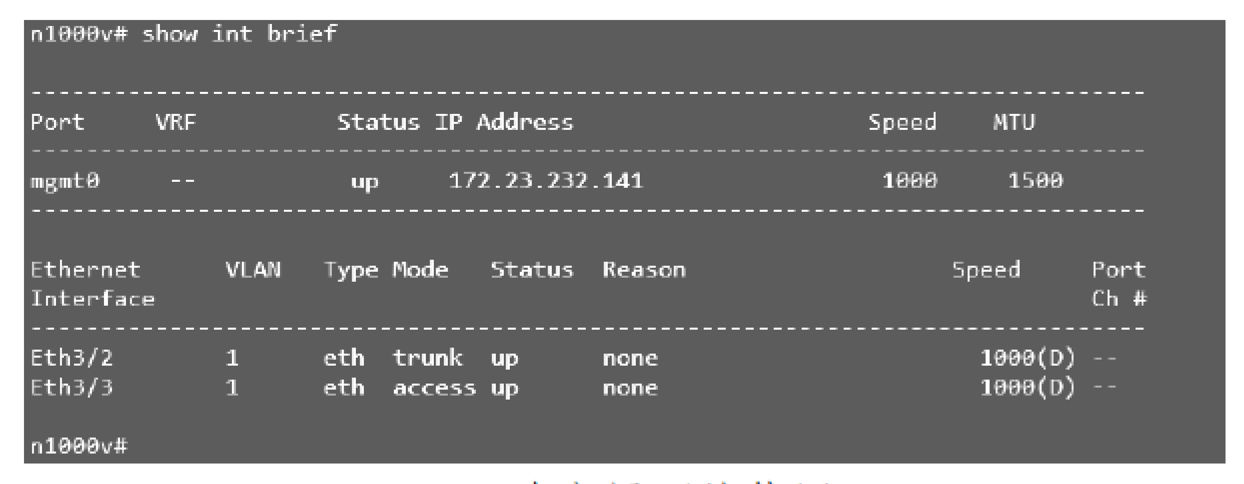
```
n1000v# show port-profile usage

-------------------------------------------------------------------------------
Port Profile                 Port        Adapter        Owner
-------------------------------------------------------------------------------
system-uplink                Eth3/2      vmnic1         172.28.30.94
vm-uplink                    Eth3/3      vmnic2         172.28.30.94

n1000v#
```

▲ 确定通信端口配置文件

9. 在 VSM 上键入 show interface brief 可以遍历界面的使用。



```
n1000v# show int brief

--------------------------------------------------------------------------------
Port      VRF           Status IP Address                        Speed    MTU
--------------------------------------------------------------------------------
mgmt0     --             up     172.23.232.141                    1000     1500
--------------------------------------------------------------------------------

Ethernet       VLAN   Type Mode   Status  Reason                    Speed     Port
Interface                                                                     Ch #
--------------------------------------------------------------------------------
Eth3/2         1      eth  trunk  up      none                       1000(D) --
Eth3/3         1      eth  access up      none                       1000(D) --

n1000v#
```

▲ 确定界面的使用

10. 当我们所有的连接落实之后，就可以在 VM 中配置网卡。选择网卡的时候，在 VM 的端口中，只要选择创建的 Cisco Nexus 1000V，并且选择你刚才创建的 data262，就可以使用 Cisco Nexus 1000V 连接了。

结 语

本章是将 Cisco Nexus 1000V 完整介绍的章节，但是在安装了 Cisco Nexus 1000V 之后才是真正的开始。虽然 Cisco 交换机的命令及操作系统并不容易学习，但在解决 VMware vSphere 的网络架构上能有相当大的帮助。Cisco 并发也推出了 Cisco Nexus 5000，是将 1000V 的功能在硬件层实现，如果你对网络的架构十分理解，相信你一定会立马舍弃 vSphere 的交换机，立即投向 Cisco Nexus 1000V 的怀抱。

读书笔记

第 24 章 使用 P2V 将物理机转换至虚拟机

关键词：

- 使用 VMware Workstation 做 P2V
- 使用 vCenter Converter 做 P2V
- 使用 True Image 做 P2V
- 使用 WinImage 做 P2V

大部分的企业在有虚拟机的产品之前就已经 IT 化了，因此不太可能创建一个完全虚拟化的环境，因此将现有的 IT 架构迁移到 vSphere 平台一直是企业最常见的做法。这个称为 P2V 的动作（Physical To Virtual），却一直是企业不敢真正放手进行虚拟化的关键之一。我们就来看看为什么。

24.1 企业虚拟化的关键

企业虚拟化最大的问题就是 P2V。P2V 最大的热点就是可用性。可用性包括了在转换后，原来服务能够继续，而用户针对服务的操作是不可以有任何改变的。我们就先来看看 P2V 的几个热点。

24.1.1 P2V 的热点

整体来说，P2V 最大的热点就是服务的延续以及在线服务的不中断。这两个热点决定了 P2V 是否成功。

1. 服务的延续

大部分的企业在进行 P2V 时，会有一个缓冲时期，会让物理机和虚拟机的服务共存一段时间，而 IT 人员则在这个缓冲期间进行系统的校调，直到整个服务都完全和物理机时一样，再将物理机正规关闭。在服务延续的部分，在 P2V 时有几个步骤及热点。

▶ P2V 延续服务的热点

- 通常会创建一个新的 IP 网段给虚拟机环境使用。
- 使用工具将物理机的硬盘转换成 VMDK。
- 如果物理机使用 HBA 连到外部存储的 LUN，则建议转换时使用 RDM 模式。
- 修改虚拟机的 VMX 以匹配物理机的需求。
- 将虚拟机引导，并且遍历各种服务的可用性。
- 进行 IP、MAC、存储的连通性。
- 让两个系统并发进行。
- 当系统常规运行一段时间之后，可将物理机上的服务停止。
- 让虚拟机运行，但要随时注意是否有问题。
- 当虚拟机运行一段时间常规后，则虚拟化落实。

之后的副本、高可用性、高扩展性，都使用虚拟机的功能，如 VMware HA、VMotion、FT、VCB/VDR 等，而不再使用物理机或网络设备的功能。

2. 上线中服务的转换

P2V 最大的问题就是服务不可以中断。许多厂家常常认为在 P2V 的过程，只要挑半夜人少的时候就行。但以网站公司来说，半夜也是服务不可以中断的时间，而跨国公司更是 24 小时上线，没有尖峰或离峰时间的分别，因此如何在服务完全不中断的情况之下，能将物理机转上虚拟机，不但是 VMware 在努力的方向，更是许多系统厂家努力的方向，而大部分的磁盘工具厂家，也一直在推出产品来帮助企业进行在线服务的 P2V。



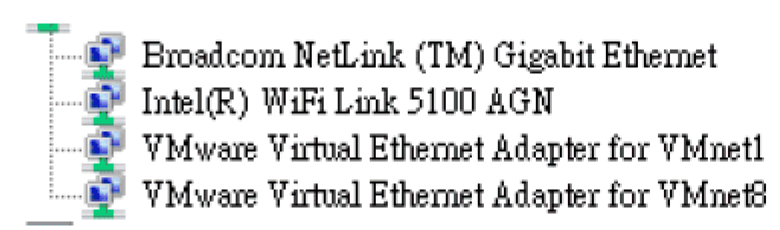
▲ 以商业网站来说，已经是 7×24 的服务时间，无法停止服务

24.1.2 P2V 常遇到的问题及解决预防

P2V 的过程中会遇到的问题千奇百怪，这是因为每一个企业对计算机的配置及设备都有所差别，因此在真正开始 P2V 之前，我们可以先做一些准备，确保在 P2V 的过程中，不会因为一些小地方而导致失败。

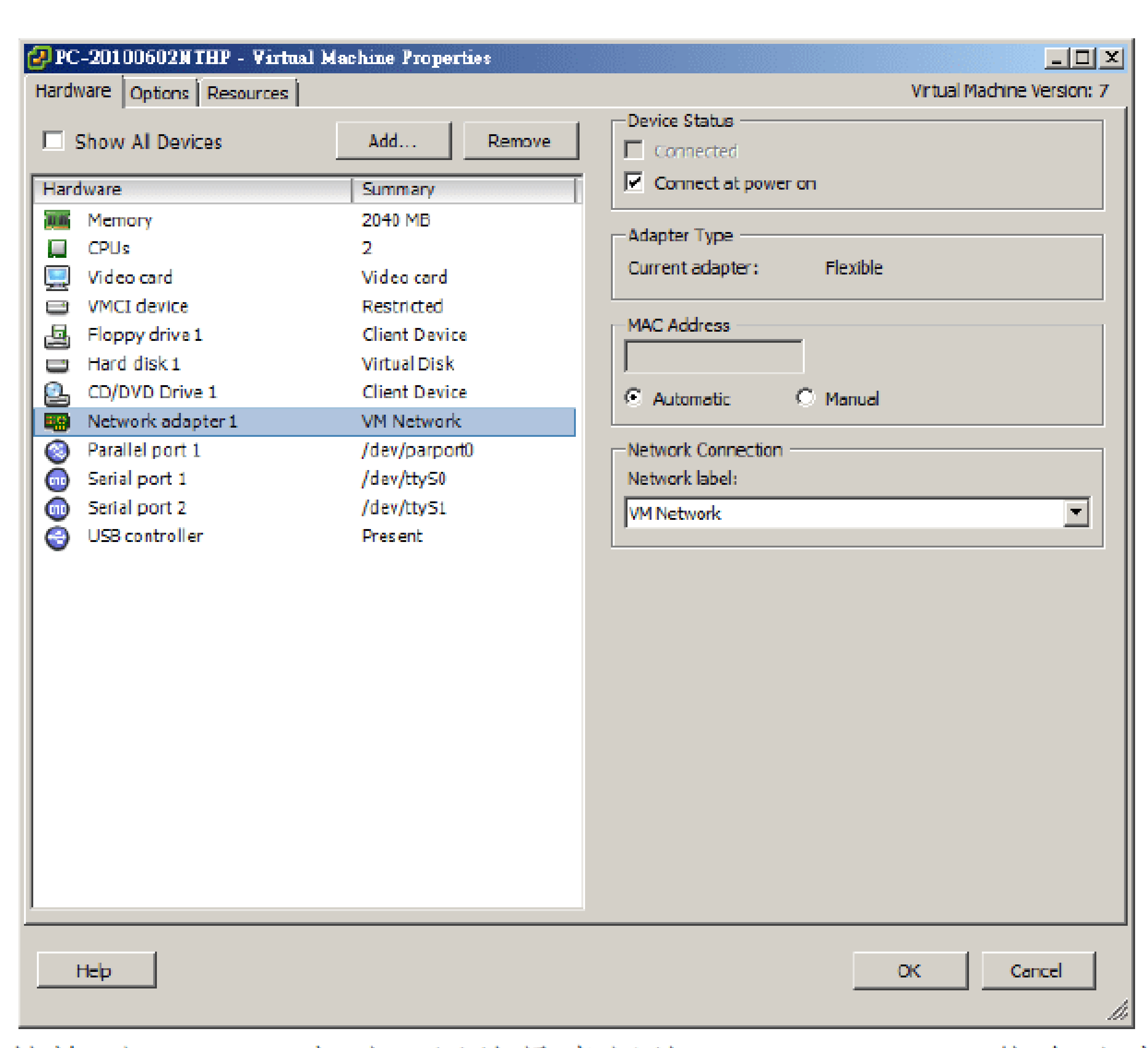
1. 网卡的问题

P2V 的过程失败，很多时候就是网卡的驱动程序在转换时失败。由于物理机使用的网卡有各式各样的品牌，但虚拟机上的选择较少，因此如果你的物理机服务器上有使用特殊网卡上的特殊功能，那么一定要试着将这个功能卸除，并且还能延续常规的服务。



▲ 在物理机上使用的是物理网卡

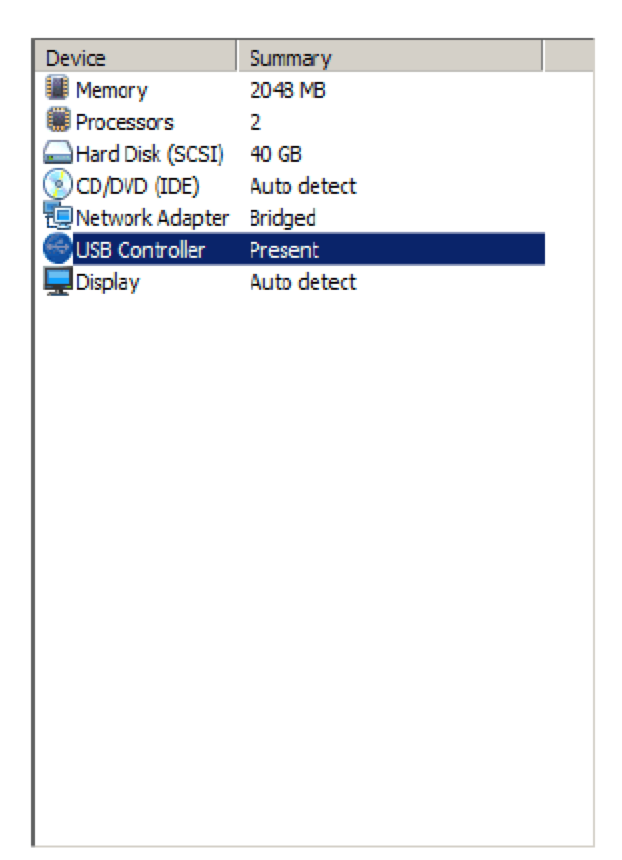
一般来说，如果在 P2V 时发现网卡的转换弹出了问题，最好的解决手段就是在 P2V 之前，将物理机的网卡拔除，并且将网卡的驱动程序拆除，并且找一片较为通用的网卡插上，以方便进行 P2V。



▲ 当转换到 vSphere 上时，用的是虚拟的 Intel E1000，可能会造成问题

2. USB 设备的问题

USB 设备也是 P2V 会出问题的地方。一般来说，如果不是非要不可的 USB 设备，通常在 P2V 之前卸除，然后在 P2V 落实之后，再将 USB 的设备插上，让虚拟机中的 USB 设备来辨认出。如果是非要不可的 USB 设备，建议在 P2V 之前，试着能否将该设备先转移到硬盘或是不需 USB 界面的设备中。



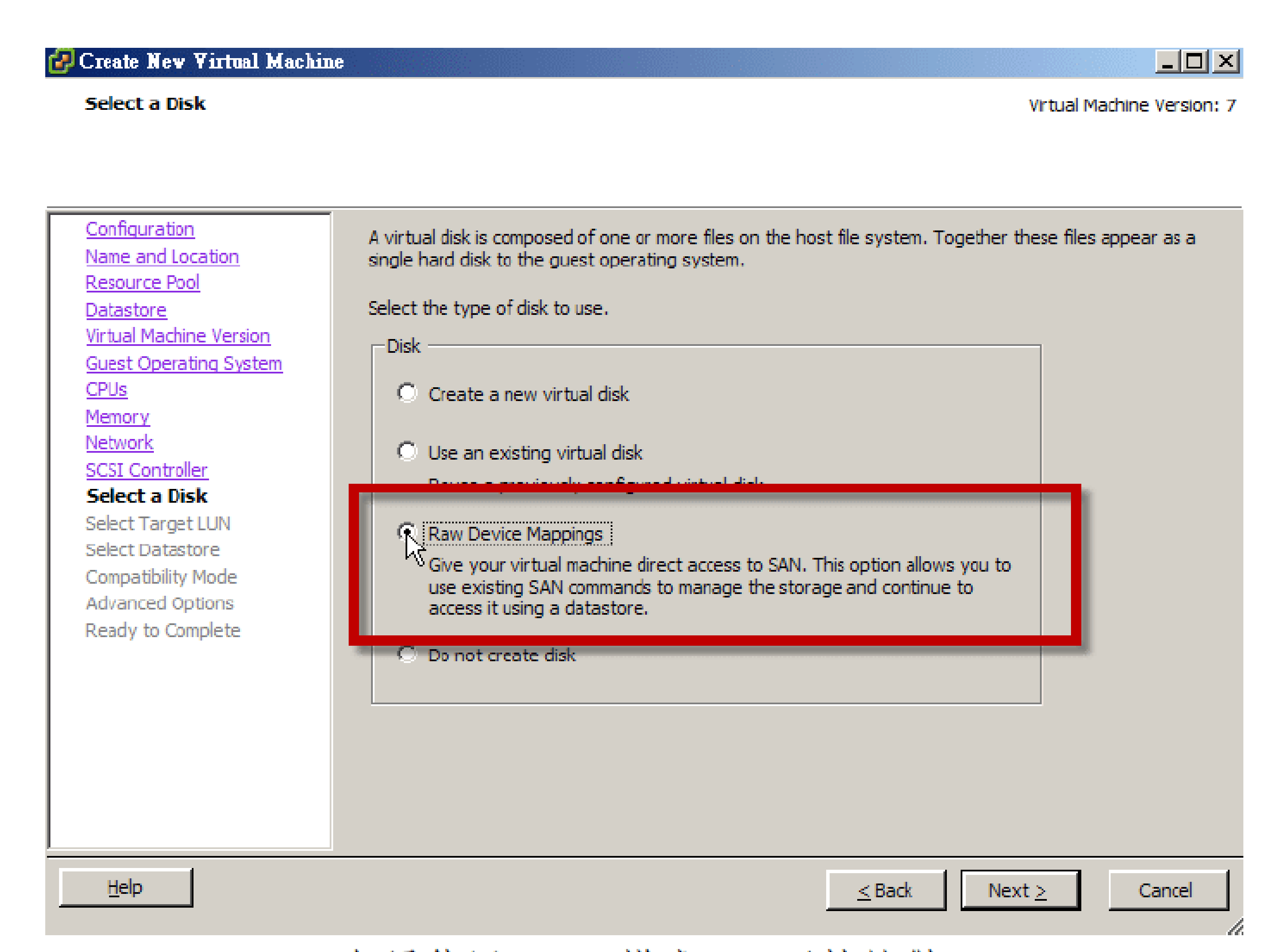
▲ VMware 的产品对 USB 的兼容性一直处理得很好，图为 Workstation

3. 使用标准的硬件运行 vSphere

在 P2V 的转换时，也常常弹出是目的端硬设备的问题。因此在 P2V 之前，目的端安装的 vSphere，一定要确定是匹配 HCL 的产品。最好不要使用 White List 或是 PC 零件的硬件。也许这个硬件设备是可以运行 vSphere 的，但在 P2V 之后的 VM 可能会无法运行。

4. 存储设备的问题

大部分的服务器都是使用本地硬盘或是外部存储设备。使用本地硬盘较简单，只要转换成 VMDK 即可。就算是 vSphere 使用了外部存储，也可以将 VMDK 放在外部存储上。如果物理机本身就是使用外部存储（如 LUN 或是 HBA、SAN BOOT）的话，那么在做 P2V 时，就可以考虑使用 RDM 模式，这样可以加快 P2V 的时间，最短可以在几分钟之内就落实 P2V 的任务。



▲ 如果使用 RDM 模式，可以快捷做 P2V

5. 应用软件的位置

服务器的实用程序是十分依赖的。从操作系统层级一直到 Frame Work 等，都会相互依赖。原来在服务器上的实用程序在虚拟化之后，如果是在网络存储设备上，问题应该不大，如果是在本地硬盘上，则要考虑其效能及是否能顺利运行进行测试，必要的话可以将该应用软件进行虚拟化。



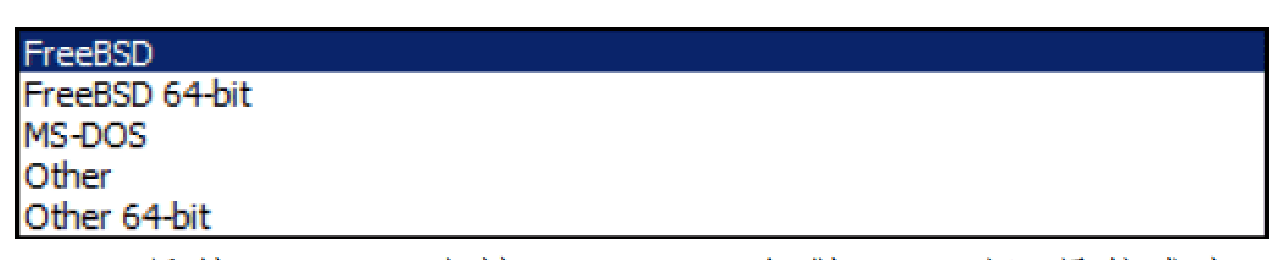
▲ 大部分的应用软件是依赖式的层级

6. 数据库的位置

数据库系统在 P2V 时是最容易出问题的。数据库通常会将数据层和逻辑层分开，举例来说，MySQL 会将运行的程序安装在一个 Linux 中，而将真正的数据库数据放在外部存储上。因此在做 P2V 时，会有许多存储设备指针的改变，会造成相当严重的问题。因此数据库在做 P2V 之前，必须先对这些文件及数据库实用程序之间进行规划。

7. 操作系统的通用性

在 P2V 时最常见的就是物理机使用的操作系统较少见。现在虽然已经是 2010 年，但有许多企业竟然还在使用 Windows NT 3.5，甚至是 MS-DOS。这是因为企业的应用环境运行稳定，并没有更换系统的必要，因此当需要更换到虚拟环境时，就会弹出这个问题。一般来说，vSphere 支持各式各样的操作系统，但较冷门的操作系统则需要在转换前进行手动配置，举例来说，FreeBSD 在进行 P2V 时，需要在 vSphere 中先创建并安装一个 FreeBSD，再将硬盘重新指向，整个动作无法使用工具落实。



▲ 虽然 VMware 支持 FreeBSD，但做 P2V 时还是较难点

8. 网络拓扑的改变

在创建虚拟环境时，为了安全起见，会先创建一个测试网段，并且让物理机和虚拟机的平台并存。此时至少 IP 会改变，因此内部的 DNS 必须重新指向。如果是使用 Windows 的 Active Directory，更需要重新加入网域。这些网络拓扑的改变，会是 IT 人员最棘手的问题，因此在真正进行 P2V 之前，必须进行完整的拓扑规划才行。

9. 老旧硬件的问题

从前的软件为了防止盗版，都会有一个保护锁，插在 Com Port 上。现在的商用软件有些也有 USB 口令锁。也有些传真服务器使用了传真卡。这一类都算是特殊的硬件，在 P2V 之后是无法使用的。任何服务都有各种不同的软件可以选择，但如果企业使用过多老旧的硬件，那么 P2V 的过程将会十分辛苦。如果是这种情况，笔者建议查找替代的服务之后再做 P2V，如果实在没有替代

的服务，可能就无法 P2V 了。



▲ 许多依赖性极高的硬件产品，就必须考虑 P2V 的可用性了

10．老旧软件的问题

老旧软件可能对虚拟机这个平台是无法适应的。尤其是以汇编语句编写的软件，是直接访问硬件的 DMA 或 IRQ 位置。这一类的软件在虚拟化之后，由于 DMA 或 IRQ 位置的改变而无法使用，因此如果在物理机是使用这一类的软件，还是需要查找其他软件来取代原来的服务，要不然也无法进行 P2V。



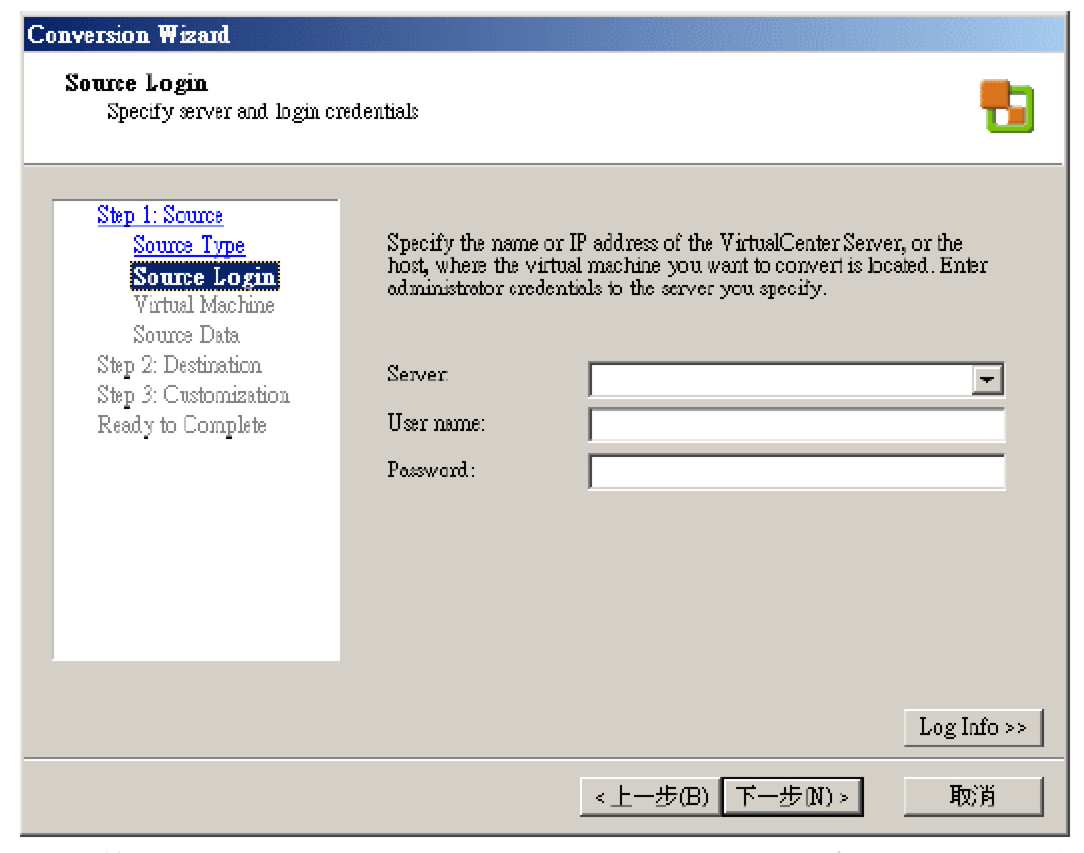
▲ 有些软件年代十分久远，但企业却用得很顺

24.2 实战 vSphere 的 P2V

将物理机转换到 vSphere 中使用的手段很多，最好用的当然是用 vSphere 自身的产品。另外第三方软件公司也有自身的工具来帮忙转换。我们在这一小节中，就来看看使用 VMware 正规的 Converter，以及使用 VMware Workstation 来转换。当然也会使用第三方的工具进行转换。

24.2.1 使用 VMware Workstation 转换

事实上，VMware Workstation 一直是 VMware vSphere 所有功能的先锋，VMware 会先将一些功能放上 Workstation 测试，当获取用户的认可或是稳定性上没问题之后，就会放到新版本的 vSphere 上。此外在 Workstation 7.0 之后，VMware Workstation 和 vSphere 的 VMDK 都可以互通，而 VMware Tools 也不需重新安装，因此我们可以使用 VMware Workstation 的功能来进行 P2V 的转换。



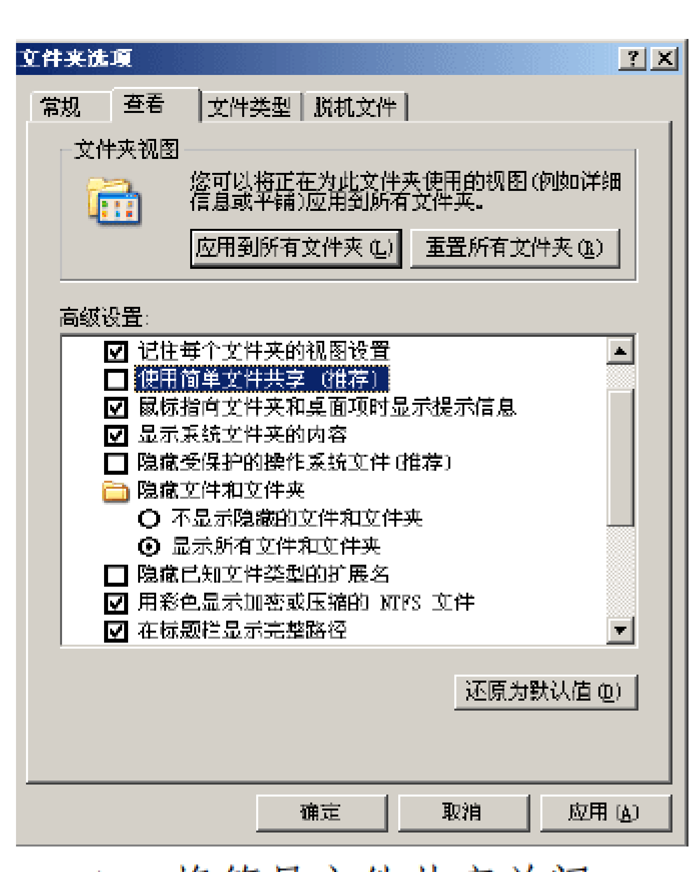
▲ 使用 VMware Workstation 中的转换功能十分方便

1. 关闭远程计算机的简易文件共享

在进行 P2V 时，常常会弹出错误消息，主要是因为 VMware 需要在被转换计算机上做些配置，将“简易文件共享”关闭。只要在要转换的计算机上，进入“我的计算机”中的“工具/文件夹选项”命令，选择“视图”选项卡，把其中的“使用简易文件共享”撤销即可。

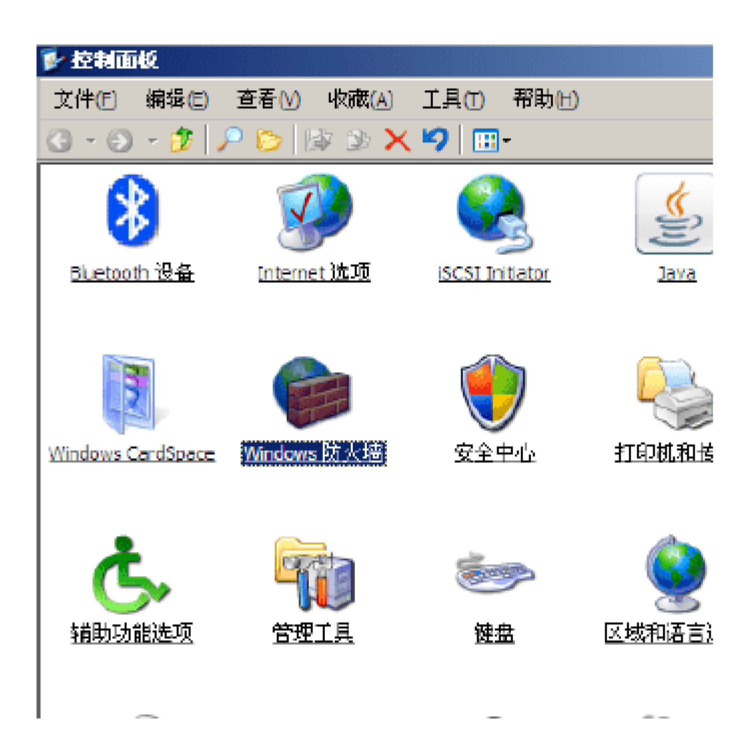


▲ 选择这里

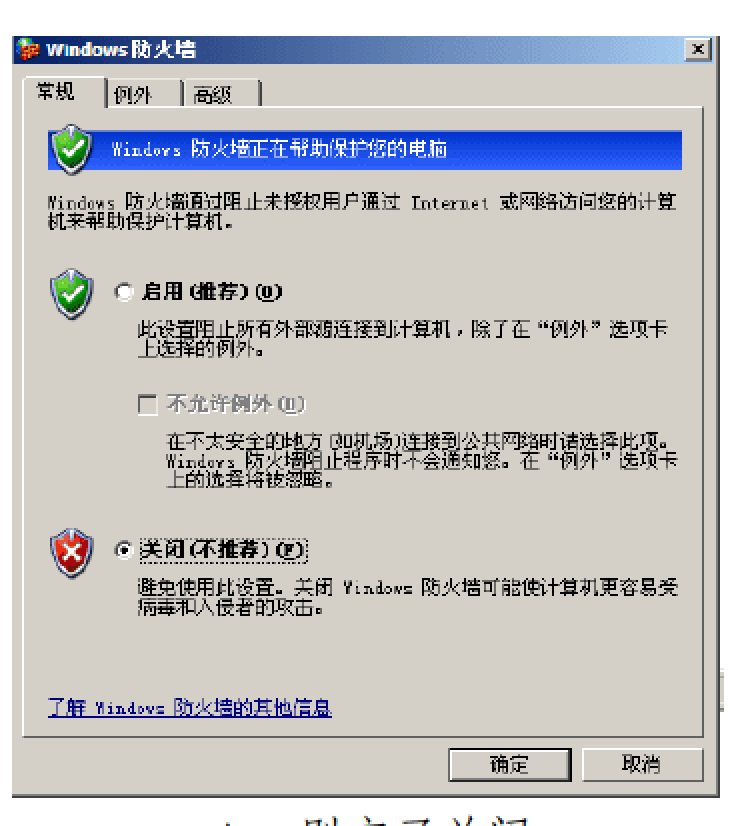


▲ 将简易文件共享关闭

此外也别忘了将远程计算机的防火墙关闭。Windows 系统都在总控台中，但关闭之后，在转换落实时别忘了再重新激活防火墙，以免发生安全上的问题。



▲ XP 总控台中的防火墙选项



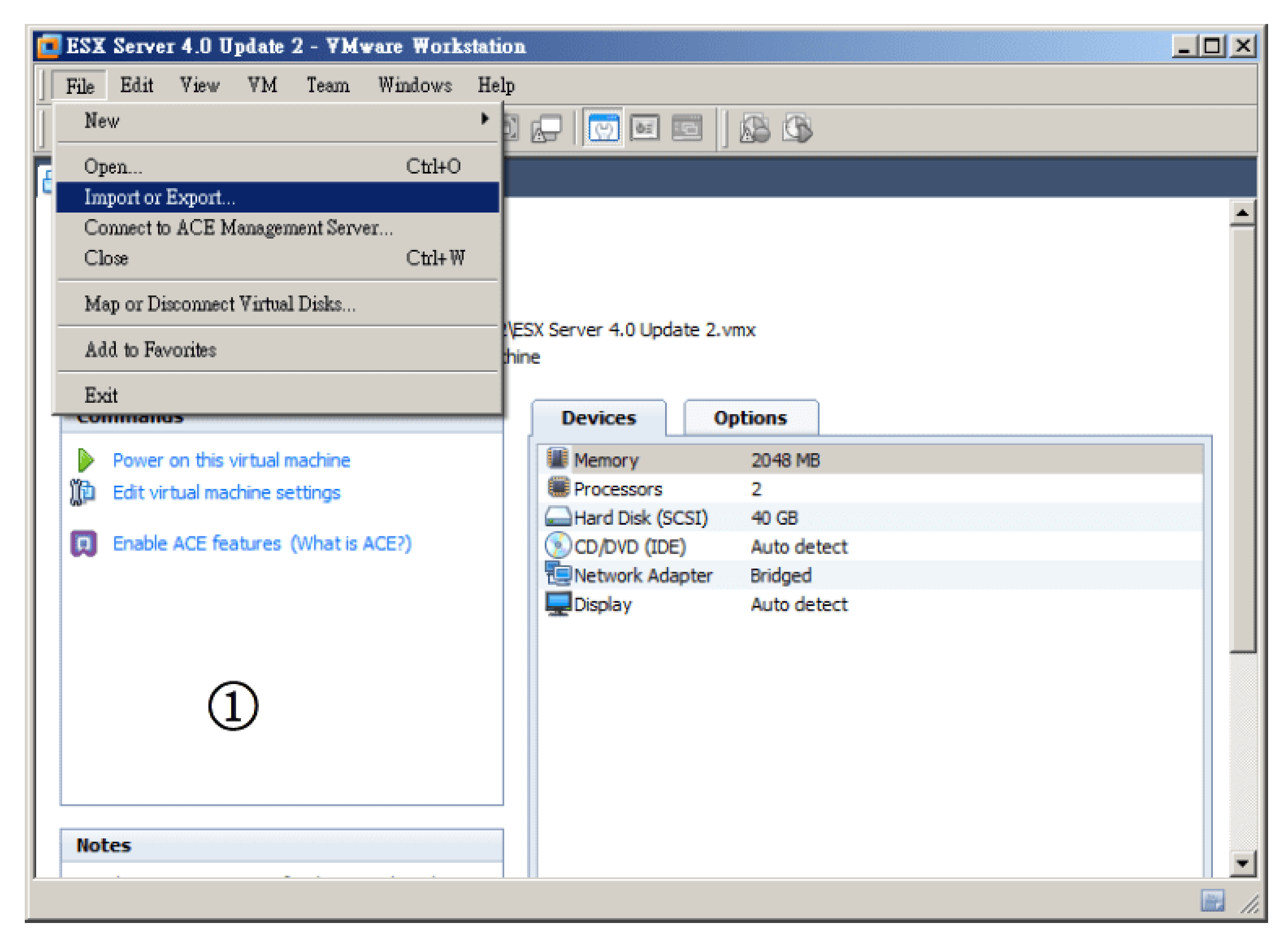
▲ 别忘了关闭

2．使用 VMware Workstation 进行转换

使用 VMware Workstation 进行转换十分简单，我们就来看看详细的步骤。

▶ 使用 VMware Workstation 进行 P2V 转换

1. 首先进入 VMware Workstation，如果是 Windows Vista/7 的话，要以系统管理者的身份进入。进入后选择 File/Import or Export 选项。

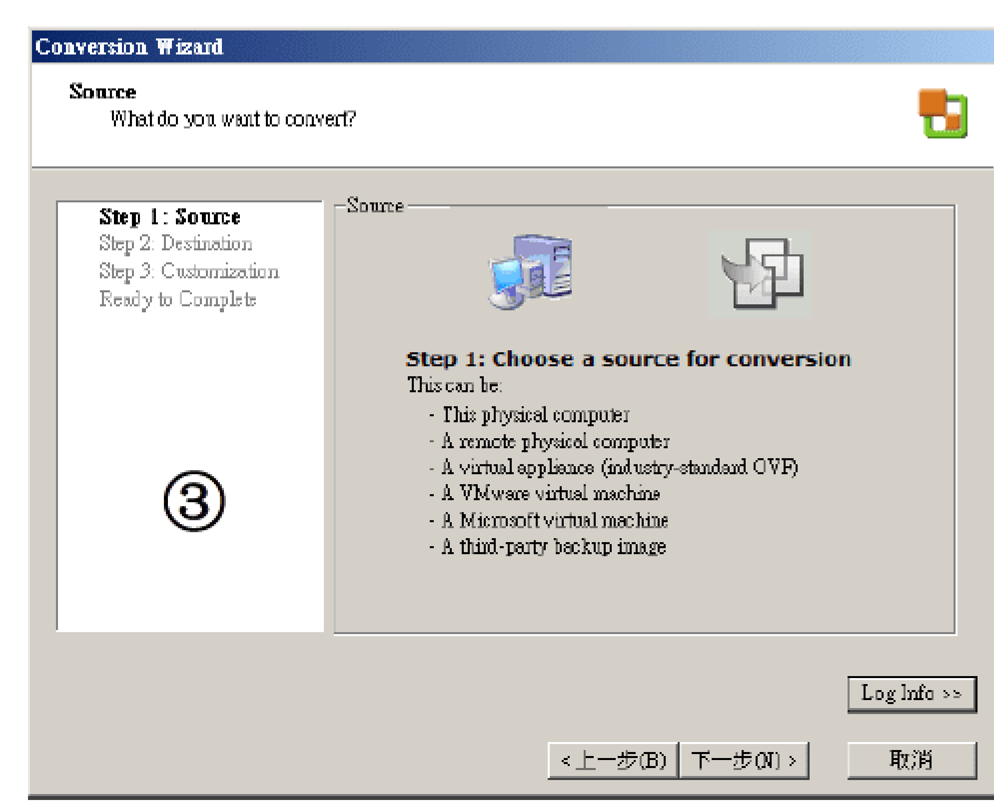


▲ 选择选项

2. 此时可以选择 P2V 或是 V2V 等功能，单击 Next 按钮继续。
3. 接下来是选择来源。也是单击 Next 按钮继续。

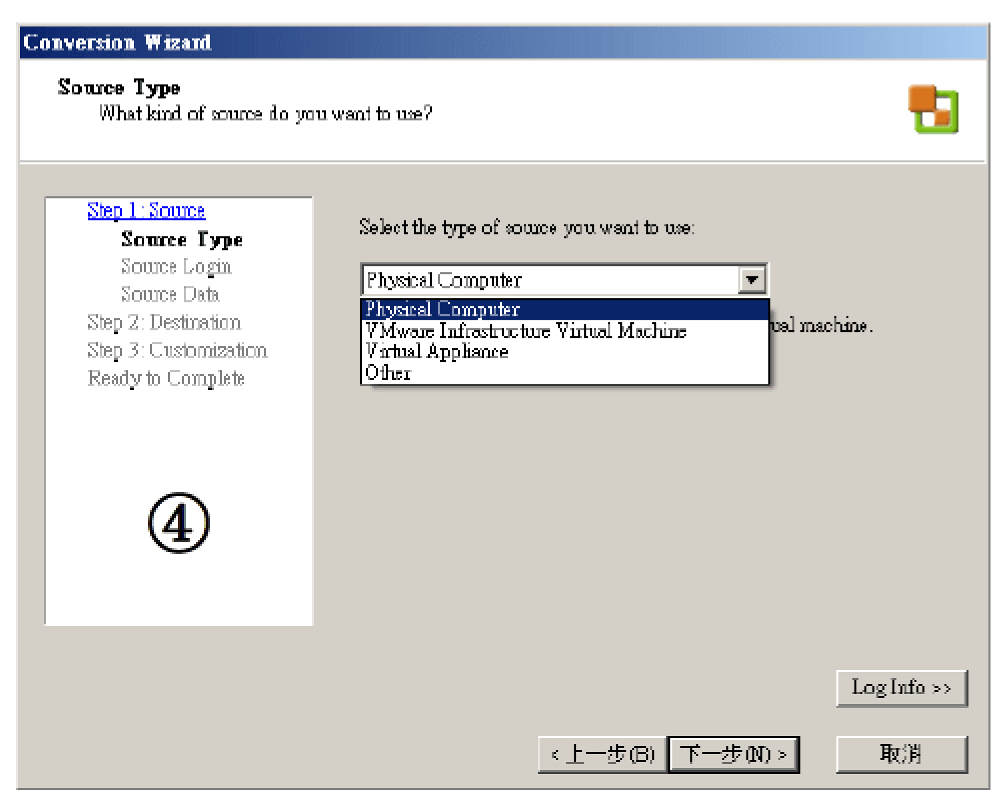


▲ 所谓的 Import 就是转换

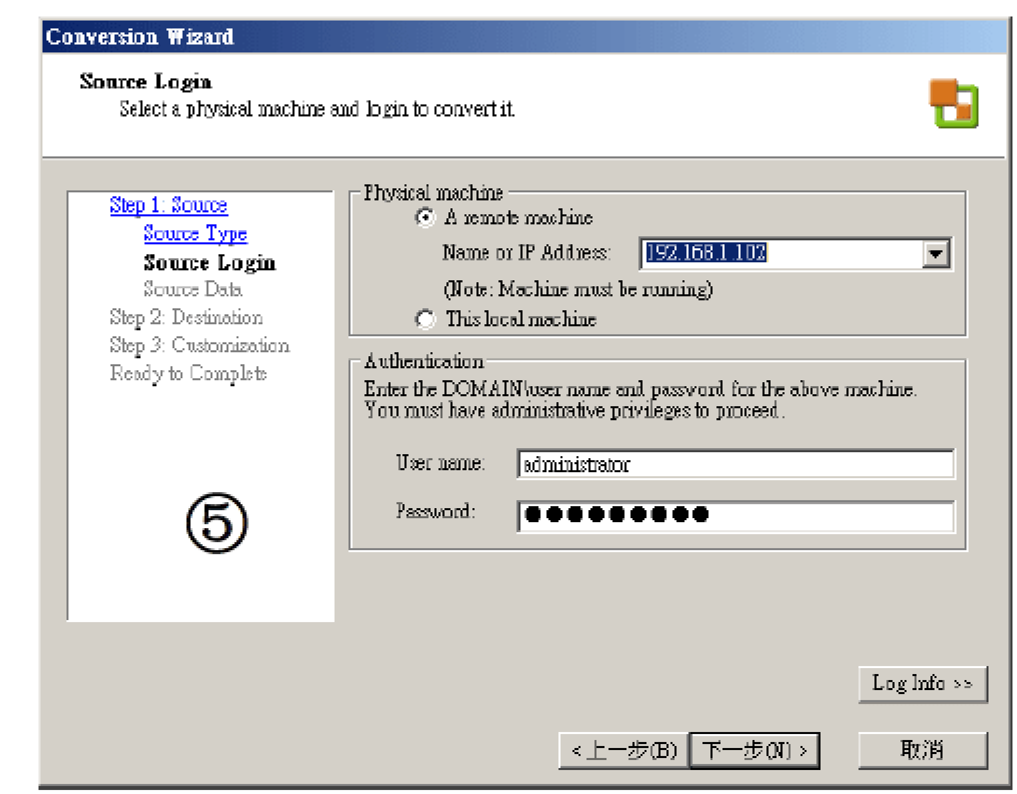


▲ 选择来源

4. 此时可以选择来源。其中有物理机、虚拟机、VA 等。我们是要进行 P2V，就选择物理机。单击 Next 按钮继续。
5. 接下来是键入物理计算机的 IP，或是你也可以转换当前使用的计算机。在键入完计算机来源之后，别忘了键入这台电脑的账号口令。注意这个账号必须是拥有系统管理员权限。我们就以一台网络上的计算机为例子。键入完之后单击 Next 按钮继续。



▲ 支持各种来源

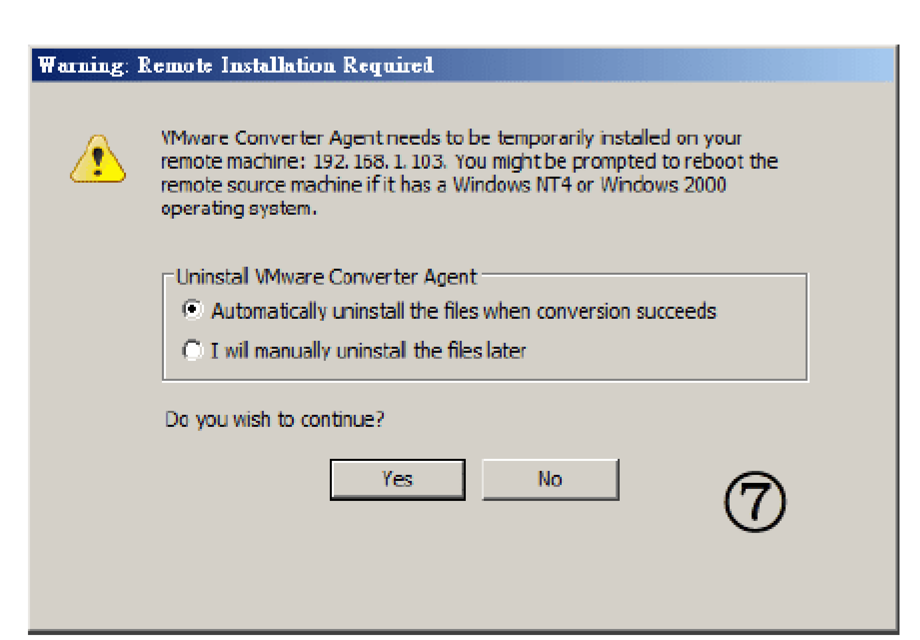


▲ 选择要转换计算机的 IP、账号及口令

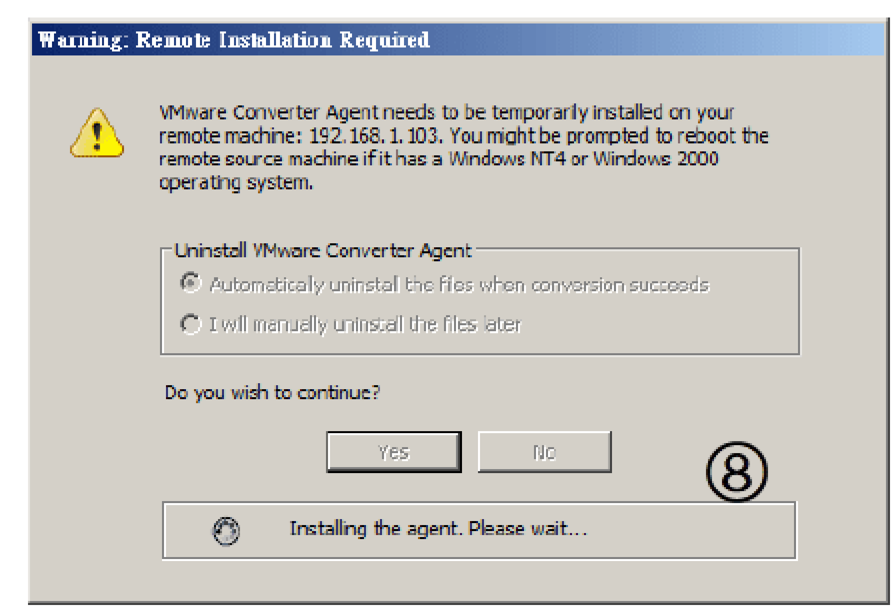
6. 此时系统会进入这台计算机，并且安装需要的软件。整个画面会成为灰色无法使用。

7. 当弹出如下图所示的画面时，意味着已经准备安装好 Converter。这个画面是让你回车是否在 P2V 之后，系统帮你拆除 Converter。这里两个选项都行。选完之后单击 Yes 按钮。

8. 接下来会真正开始安装 Converter。



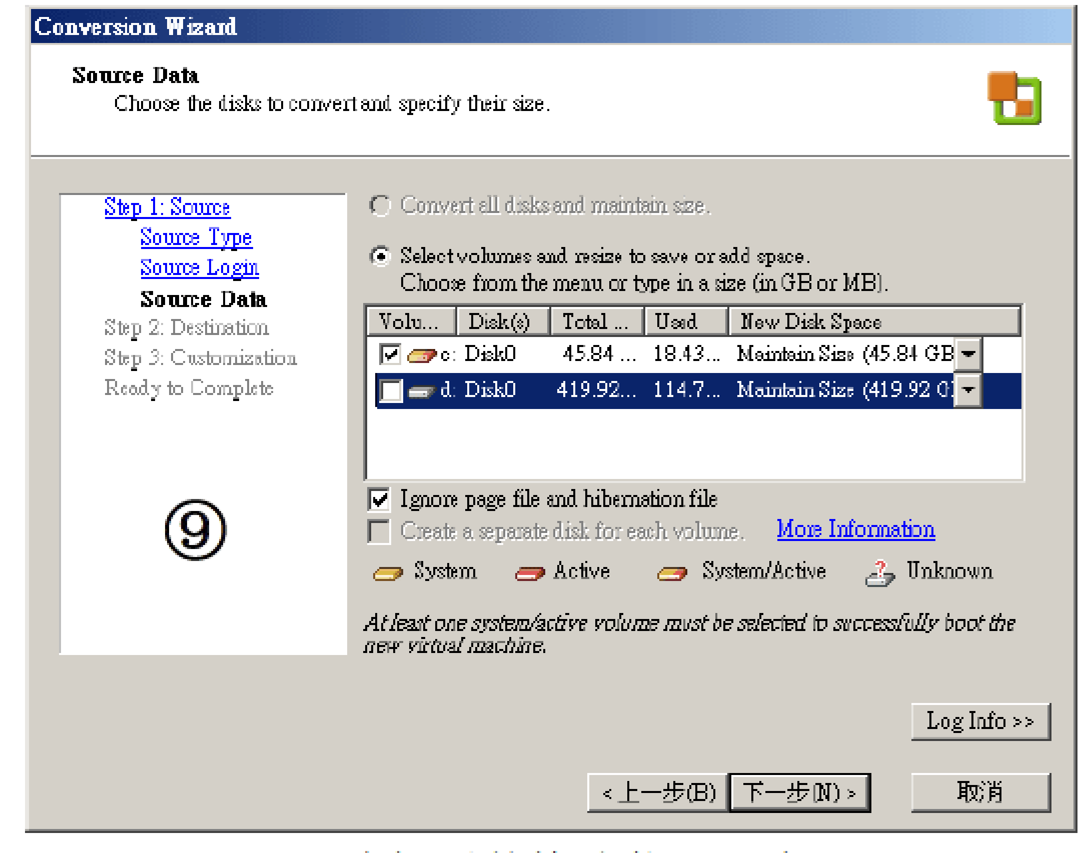
▲ 这里两个都可以选



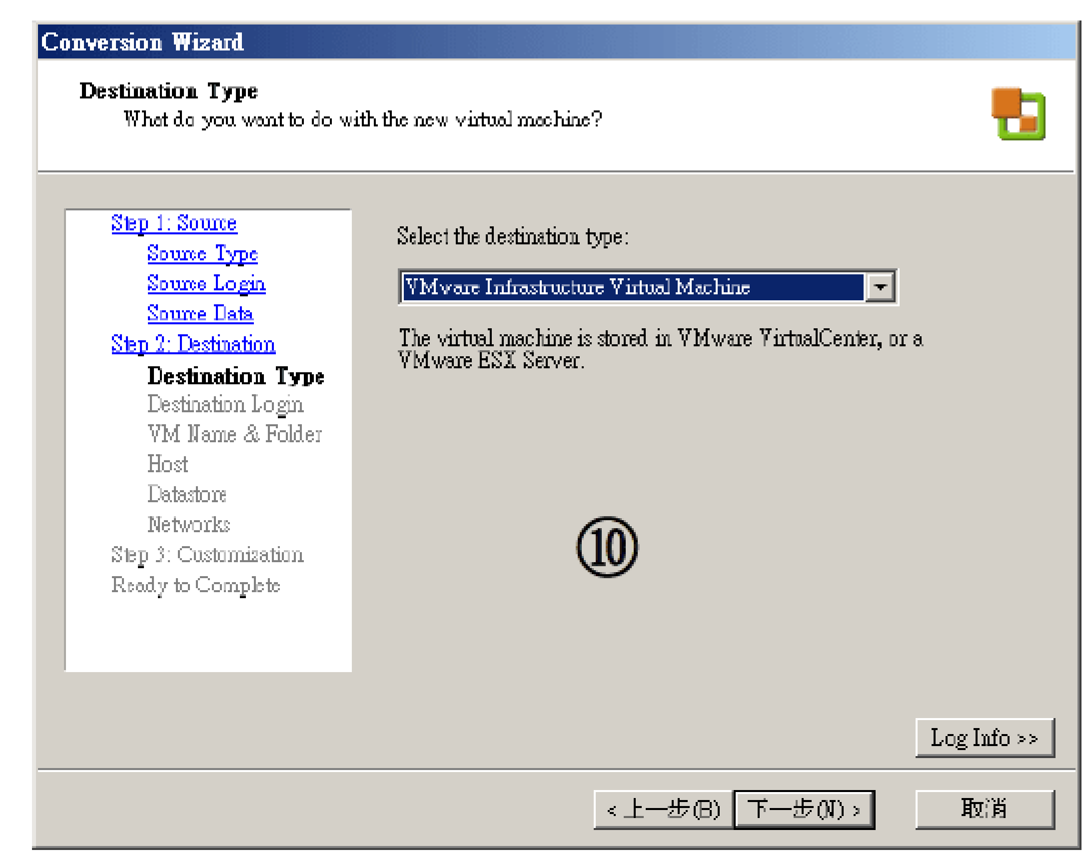
▲ 此时系统会将 VMware Converter Agent 装上物理计算机

9. 在安装好之后，系统会询问你要转换哪一个磁盘，这里为了测试目的，就不转换 D 磁盘了。在正规的 P2V 中，系统会为物理机的第二个硬盘创建一个新的 VMDK。

10. 接下来系统会询问转换的目的，单击 Next 按钮继续。

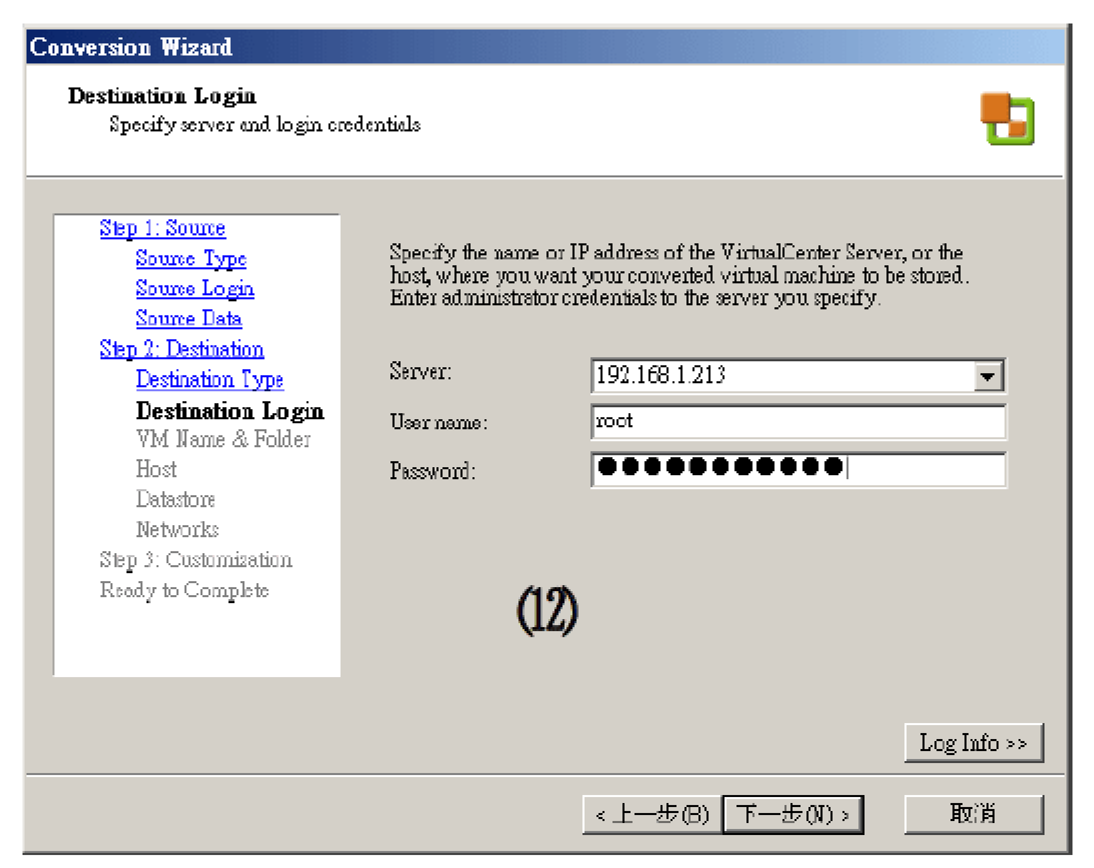


▲ 选择要转换的物理硬盘

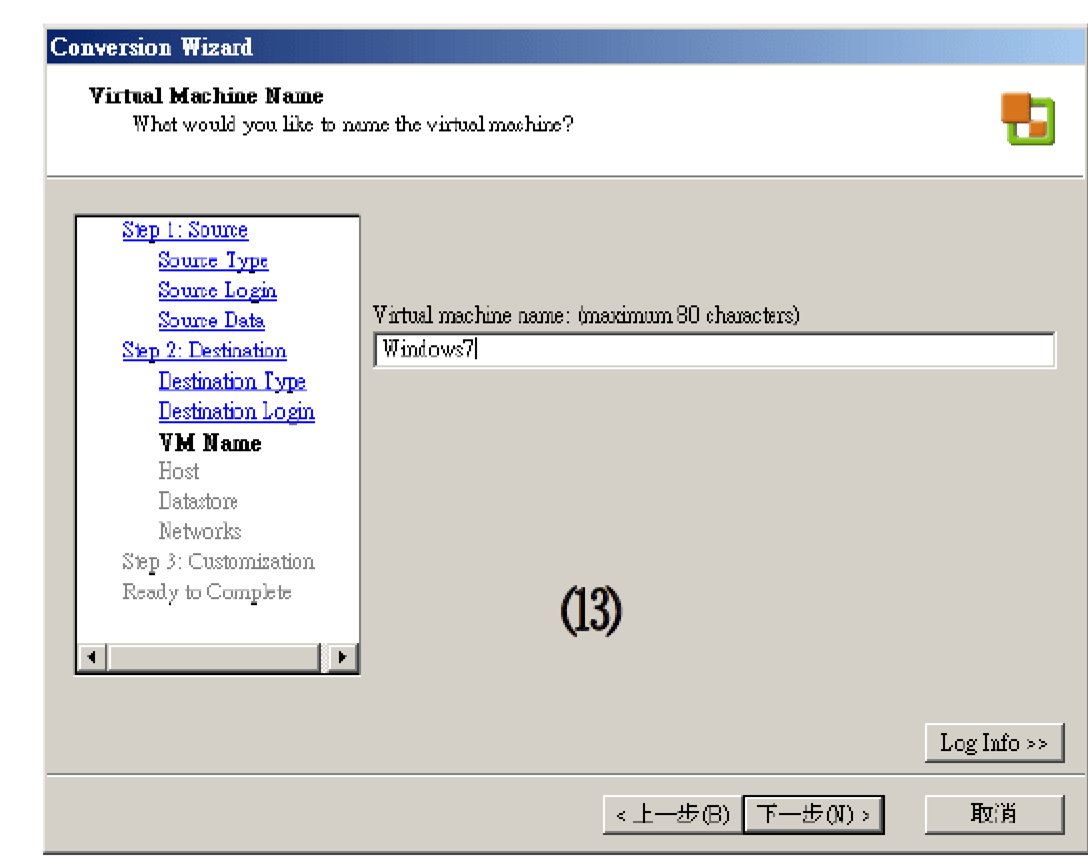


▲ 转换的目的地，当然是 vSphere

11. 如果是要转换到 vSphere 下，就选择第一项，如果是转换到 VMware 的其他产品，就选择第二项。我们选择第一项转换到 ESX 中，选择完之后就单击 Next 按钮继续。
12. 接下来会要求给定 ESX 的 IP，账号及口令，单击 Next 按钮继续。
13. 接下来键入这个 VM 的名称，我们就键入 Windows7。单击 Next 按钮继续。

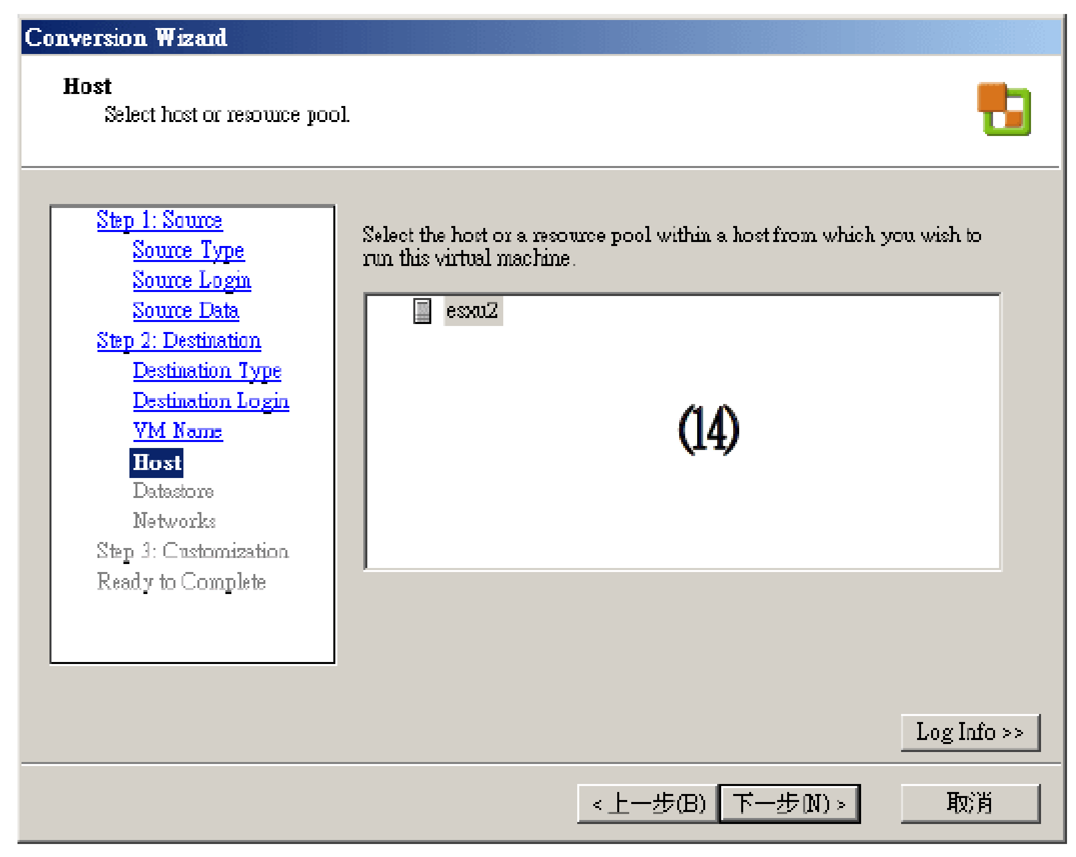


▲ 转换目的 ESX 主机的账号、口令及 IP

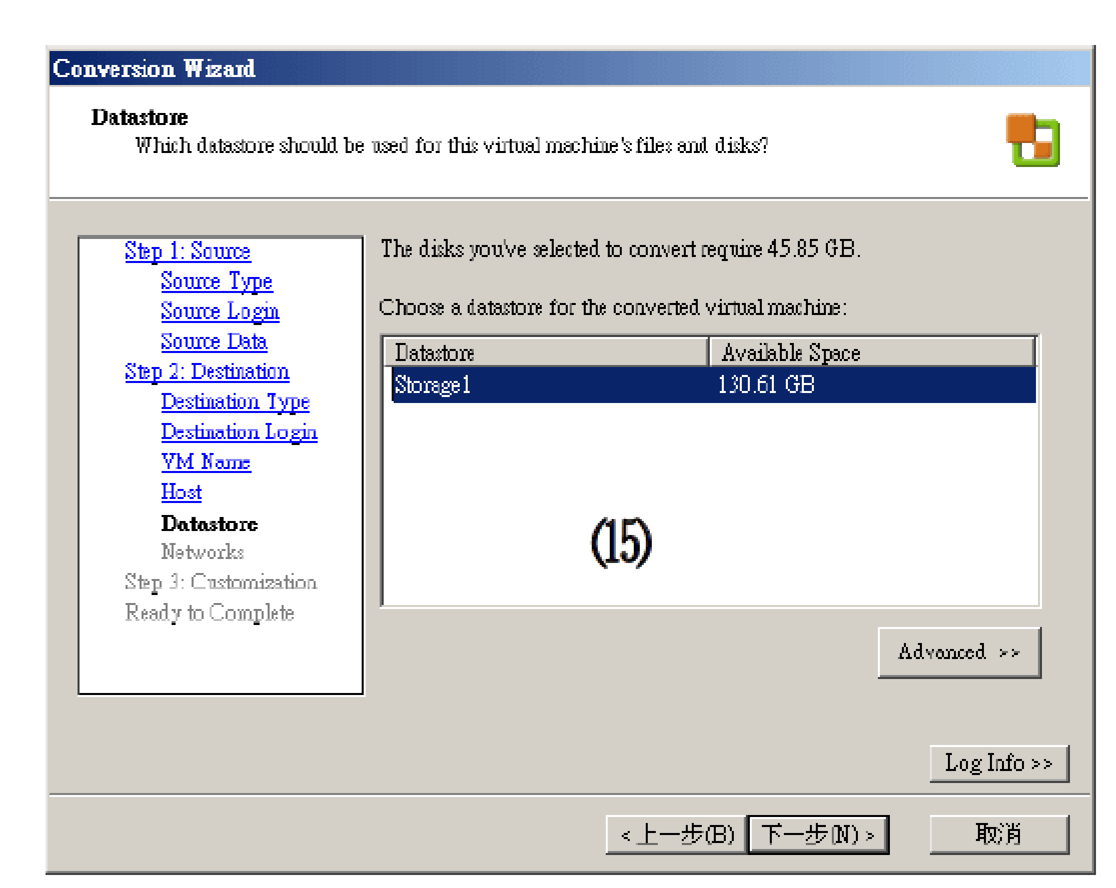


▲ 替这个 VM 取一个名称

14. 接下来是选择 Resource Pool，本示例为单机，直接选择后单击 Next 按钮继续。
15. 接下来会选择要存放的 Datastore。单击 Next 按钮继续。

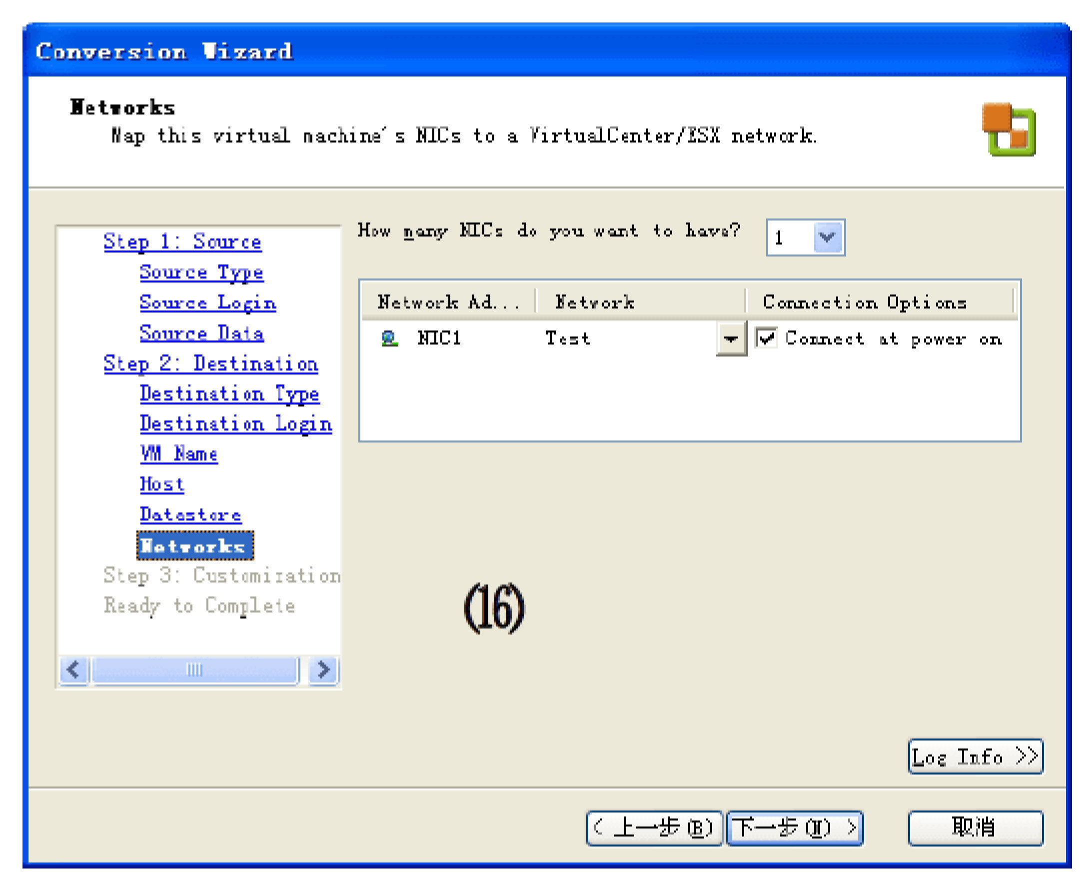


▲ 选择该 ESX 主机上的资源

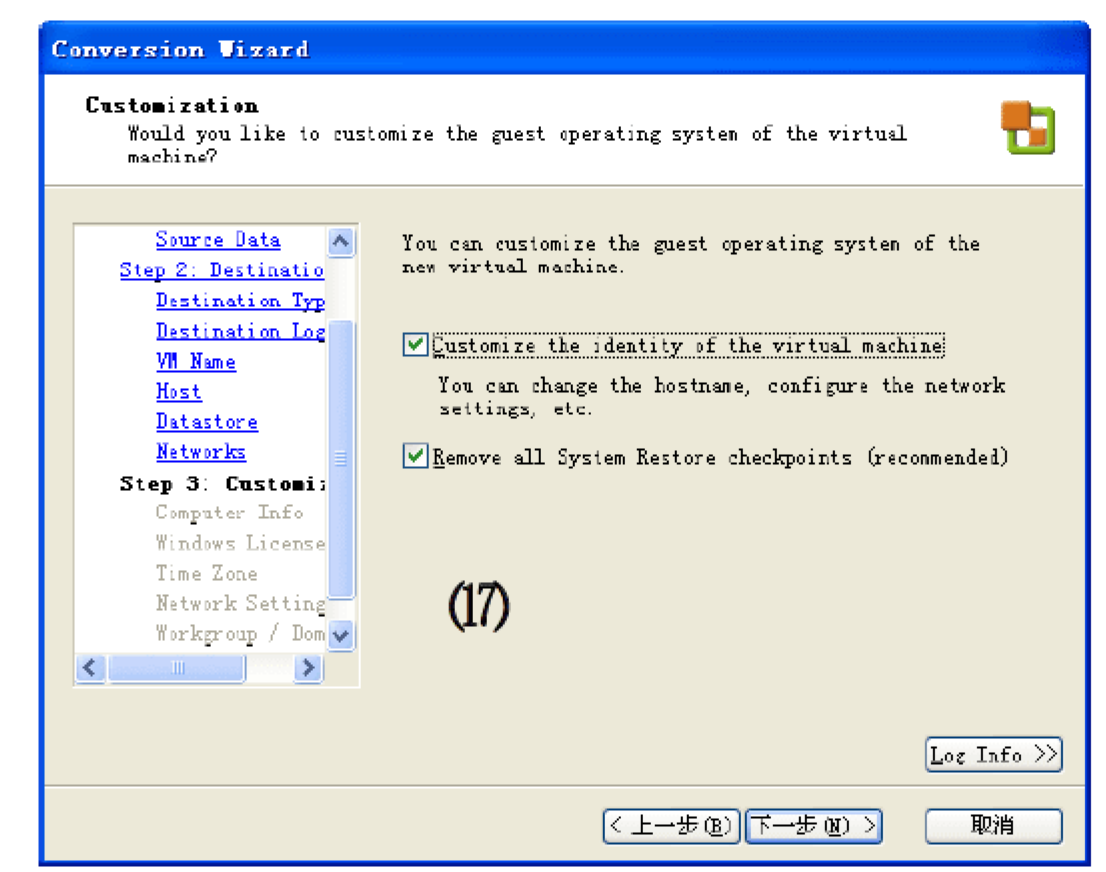


▲ 选择要存放的 Datastore

16. 接下来是选择这个主机在转换后要使用哪一个 VM 的 Port Group。单击 Next 按钮继续。
17. 接下来是选择是否要进行客制化的校调以及卸除所有的遍历点。所谓的客制化校调，是在这台物理机转换后要作为模板机时才需要。如果只是转换一台机器则没必要。单击 Next 按钮继续。



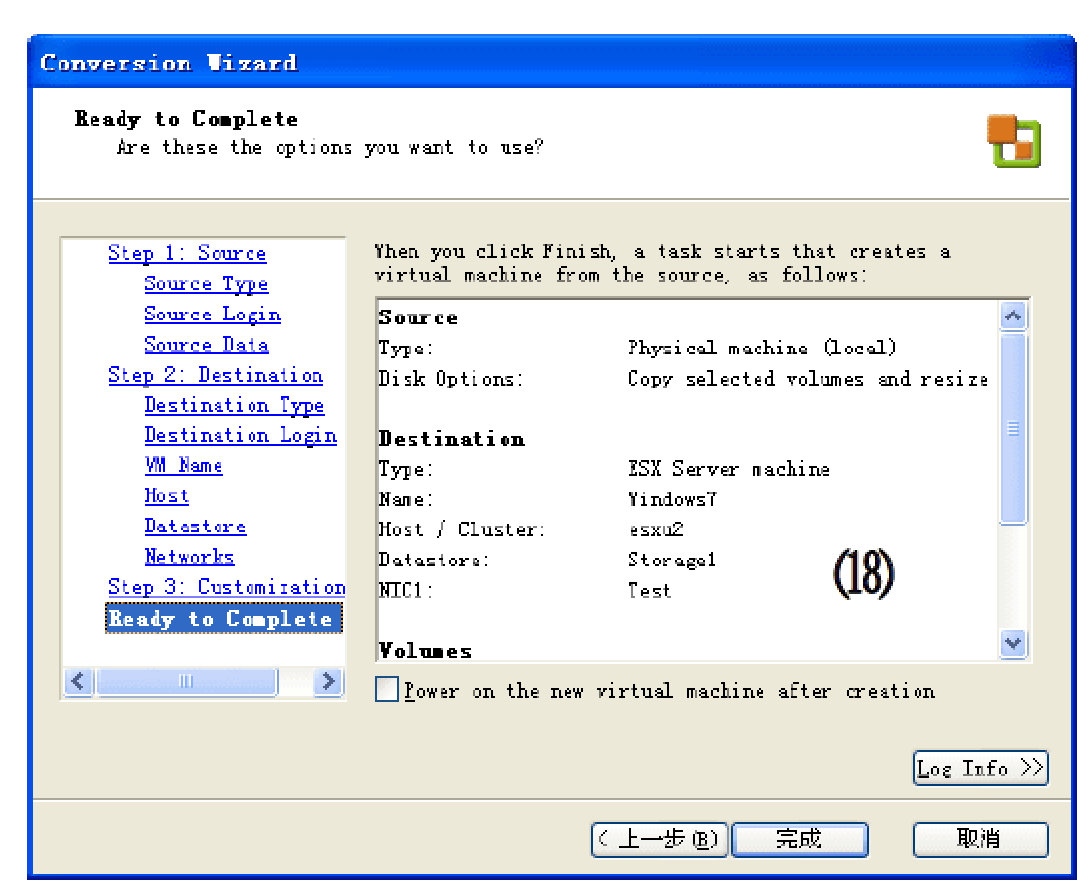
▲ 选择网络连接



▲ 转换单机时，最后一项要撤销选中，以免转换失败

18. 此时会弹出总结画面，单击 Finish 按钮则落实配置，会开始转换。

19. 当弹出如下图所示的画面时，意味着开始转换了。



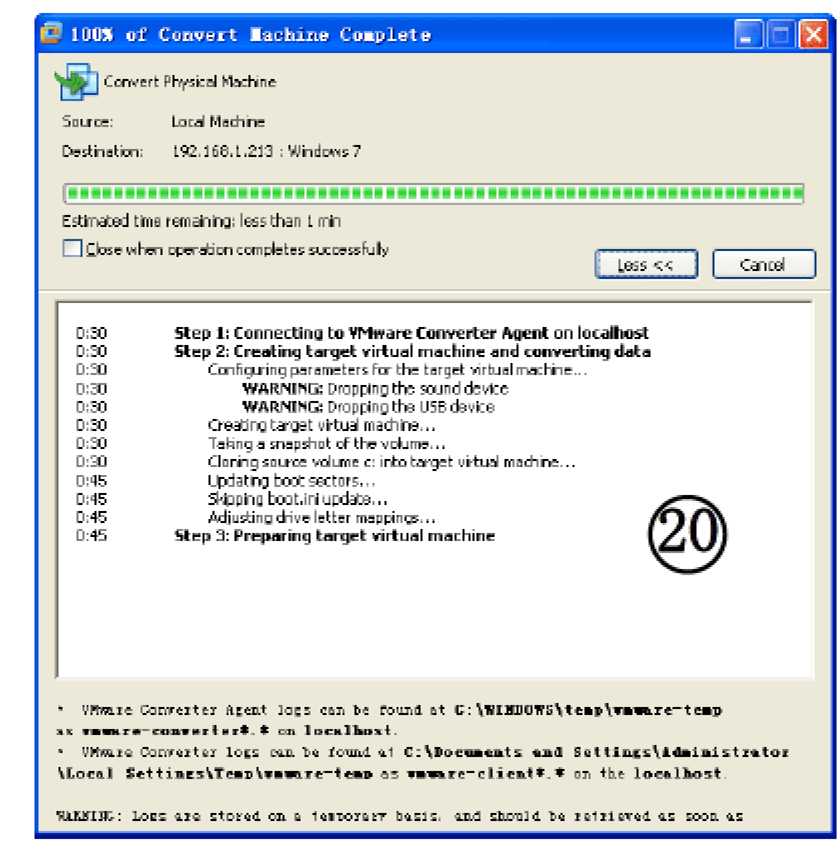
▲ 开始转换

1% of Convert Machine Complete
Convert Physical Machine
Source: Local Machine
Destination: 192.168.1.213 : Windows7
Close when operation completes successfully
Less <<
Cancel
0:22 Step 1: Connecting to VMware Converter Agent on localhost
0:22 Step 2: Creating target virtual machine and converting data
0:23 Configuring parameters for the target virtual machine...
(19)

▲ 转换的时间有时会超过几小时

20. 当弹出如下图所示的画面时，意味着转换落实。

21. 你可以进入 vSphere 中，试着将这个转换过的机器引导，看看和原来的物理机有何不同。



▲ 转换落实



▲ 弹出在 vSphere 中了，可以引导试试

24.2.2 使用 VMware 的 vConverter Standalone

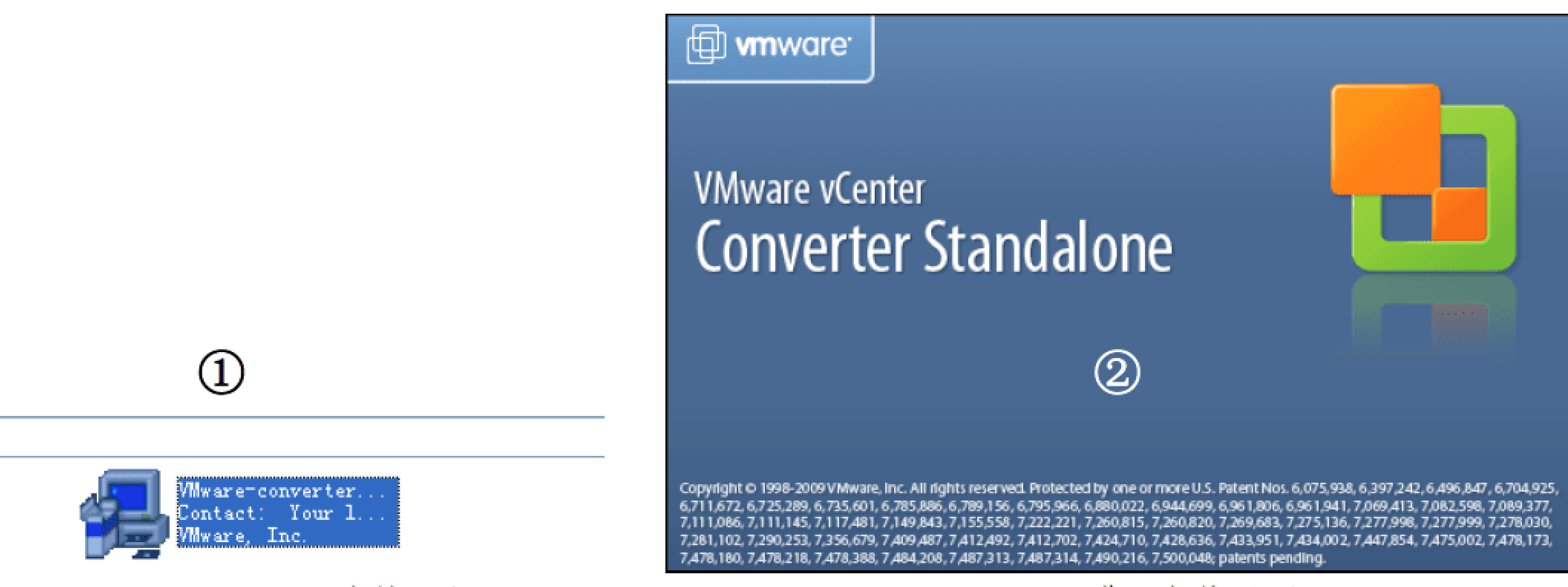
VMware 为了帮助大家顺利进行 P2V，也推出了专门进行转换的产品 VMware Converter。这个产品可以安装在任何 Windows/Linux 的计算机上，在安装好之后，就可以用这台计算机作为 Client，将网络上的任何计算机转入 ESX 主机中，我们就来看看。

1. 安装 VMware Converter

我们可以到 VMware 的网站上下载 VMware Converter，当前最新的版本是 4.0.1，下载回来安装即可。

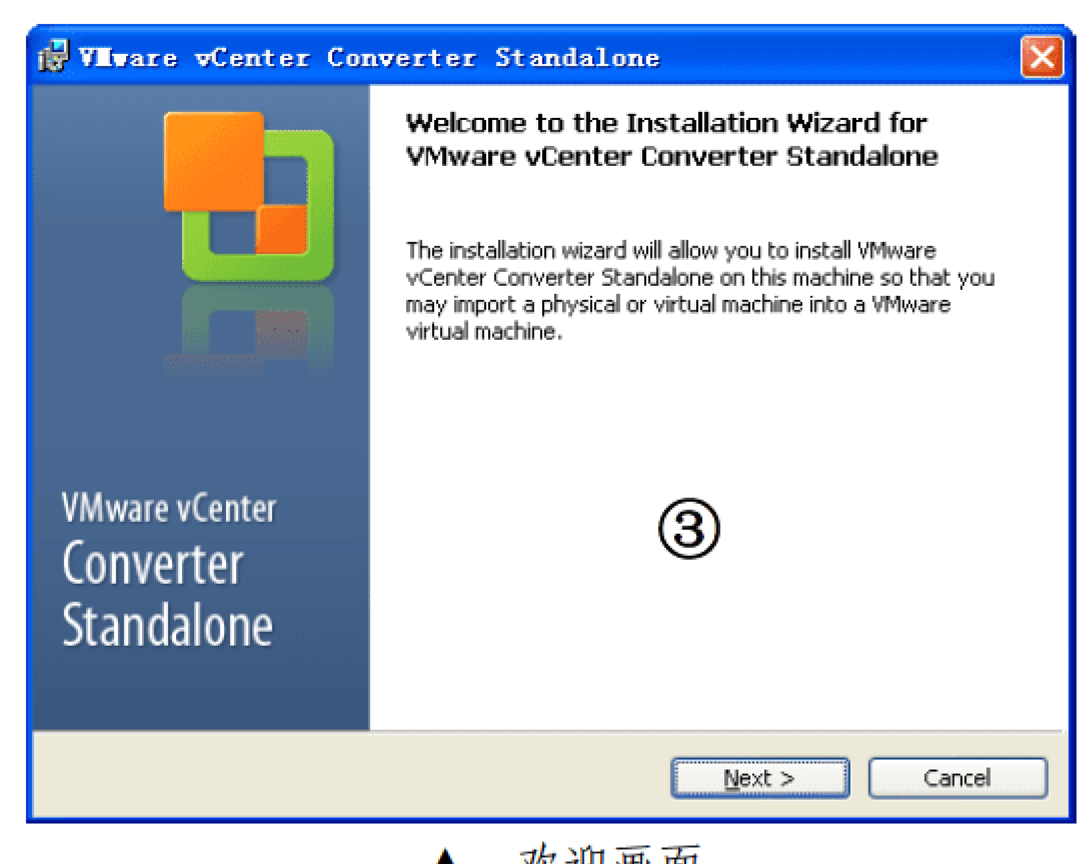
▶ 安装 VMware Conterver

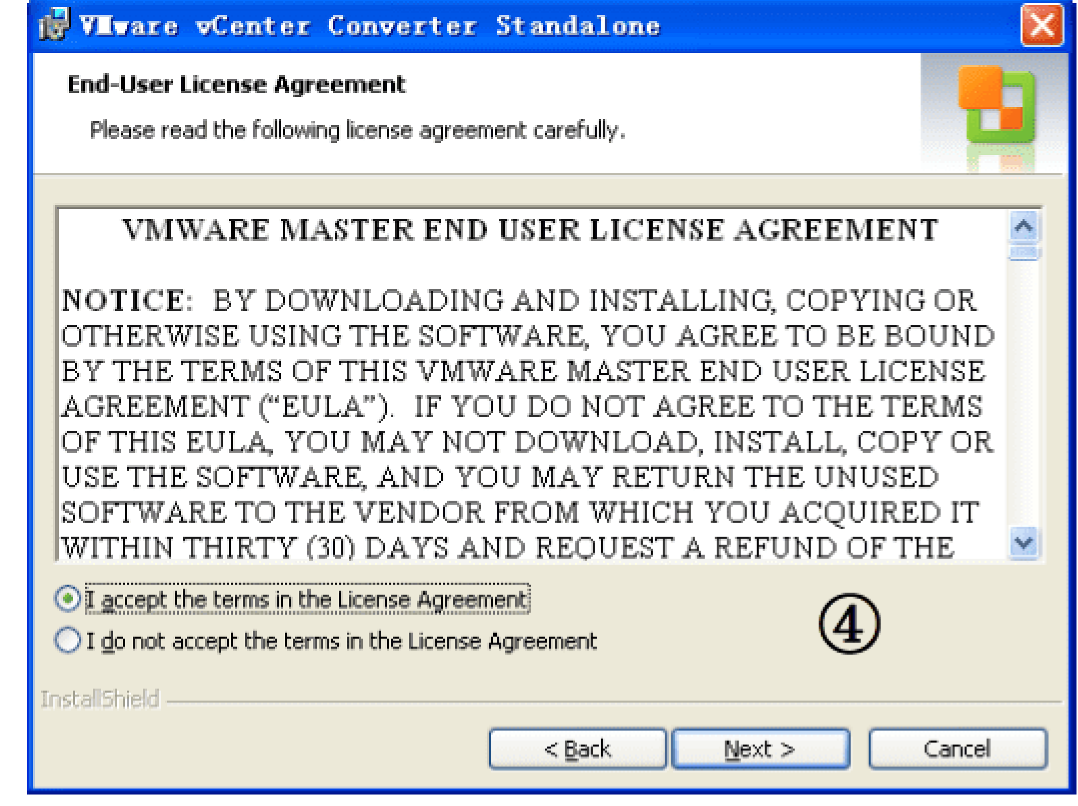
1. 首先运行下载回来的 VMware Converter。
2. 此时会进入安装画面。

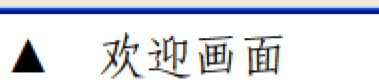


▲ 直接运行　　　　▲ 进入安装画面

3. 接下来会进入欢迎画面，单击 Next 按钮继续。
4. 此时会进入版权画面，接收后单击 Next 按钮继续。

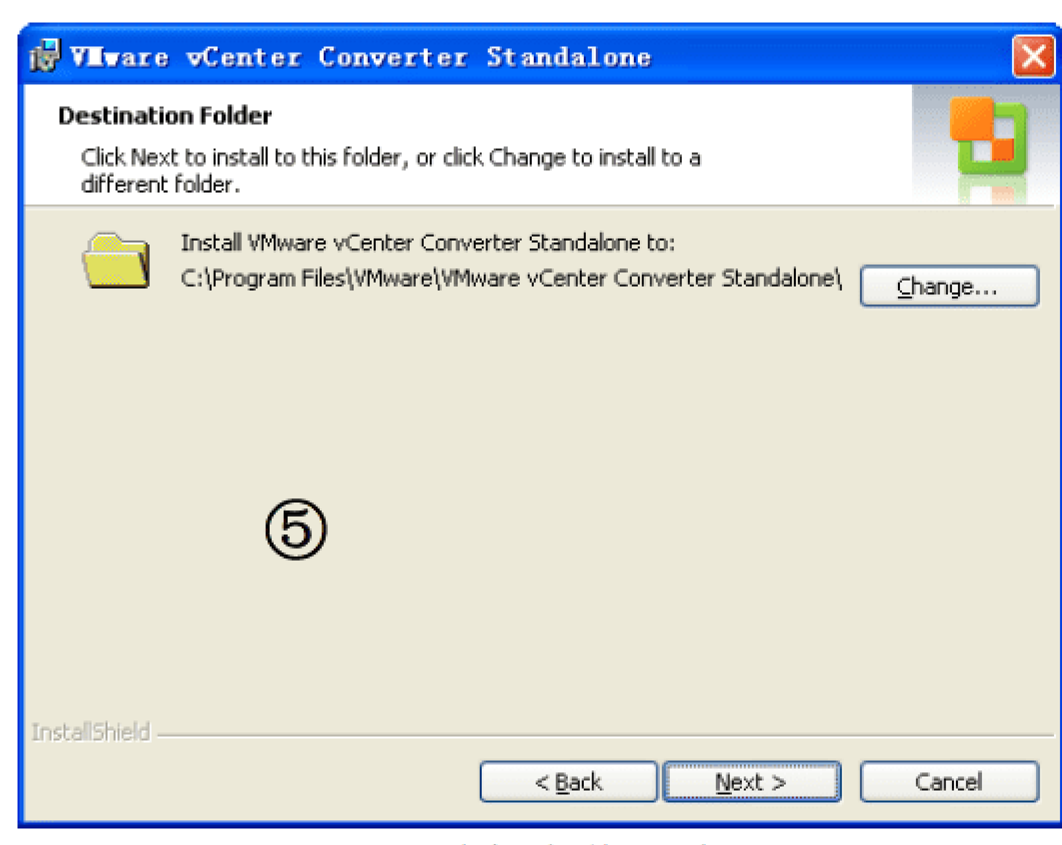




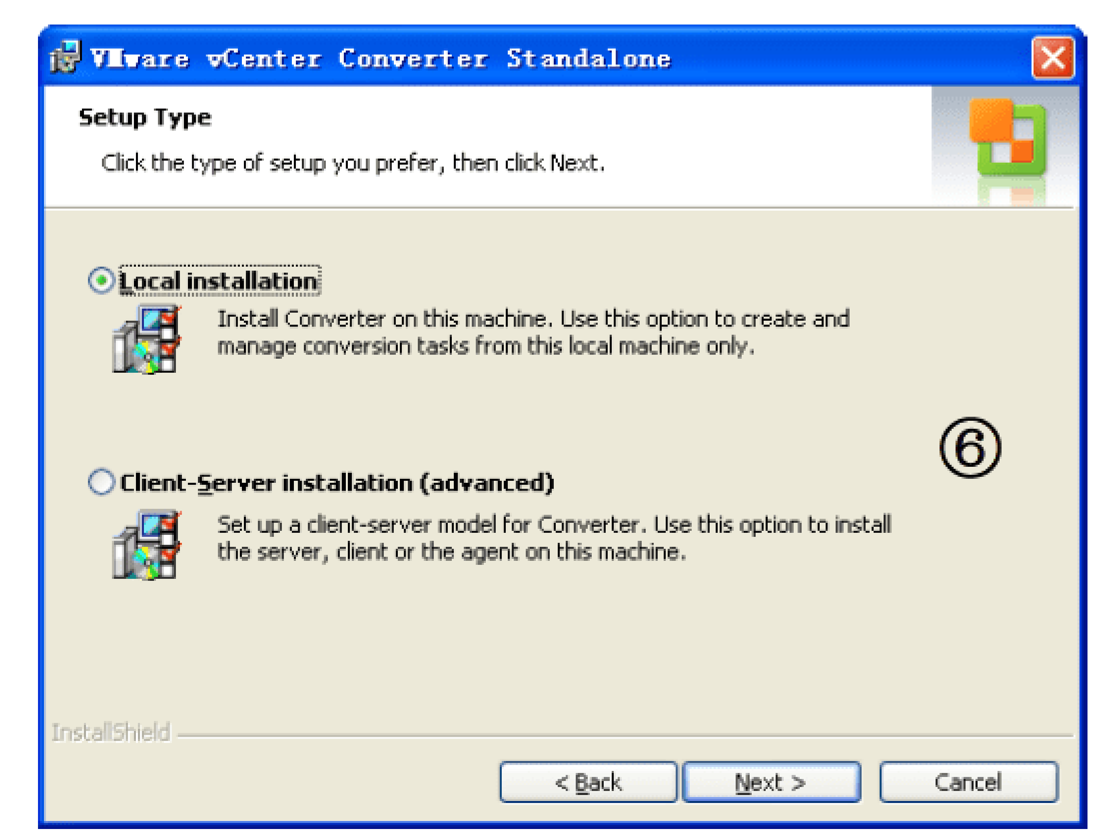


▲ 版权画面

5. 接下来选择安装目录，单击 Next 按钮继续。
6. 选择安装方式，我们使用第一种的本地安装即可。单击 Next 按钮继续。



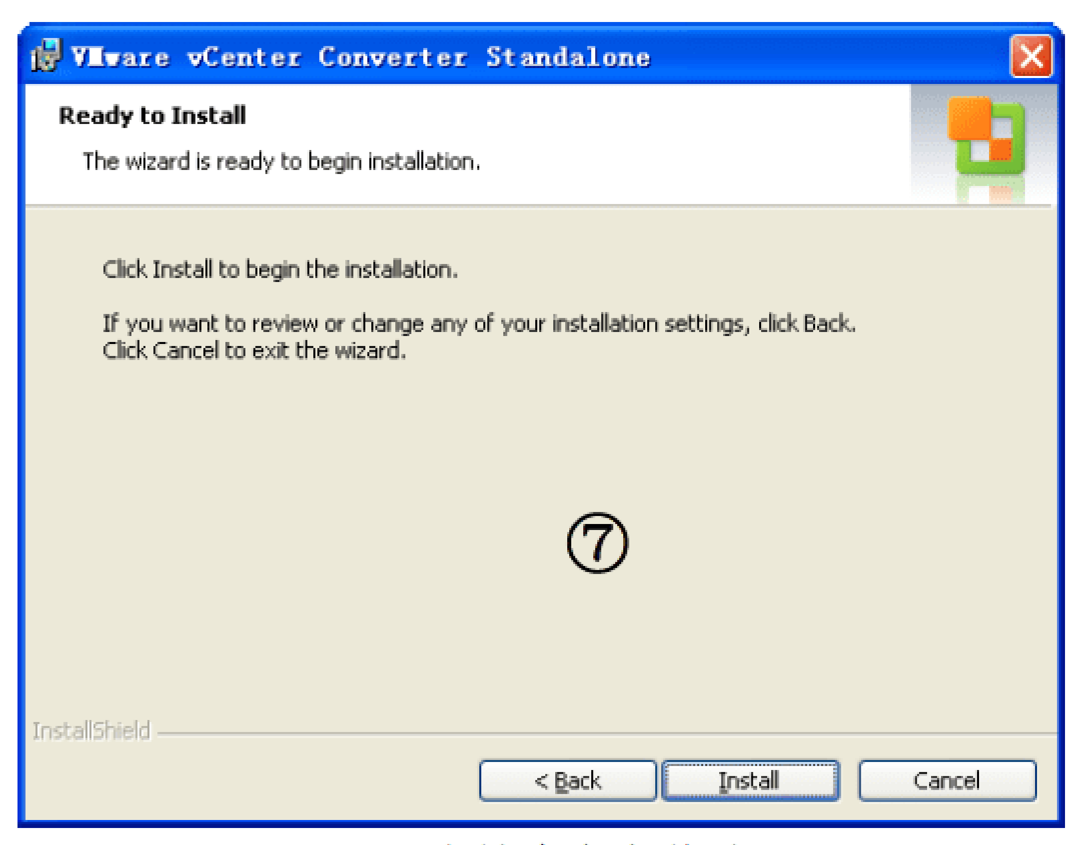
▲ 选择安装目录



▲ 选择安装方式，用本地安装即可

7. 接下来进入总结画面，单击 Install 按钮继续。

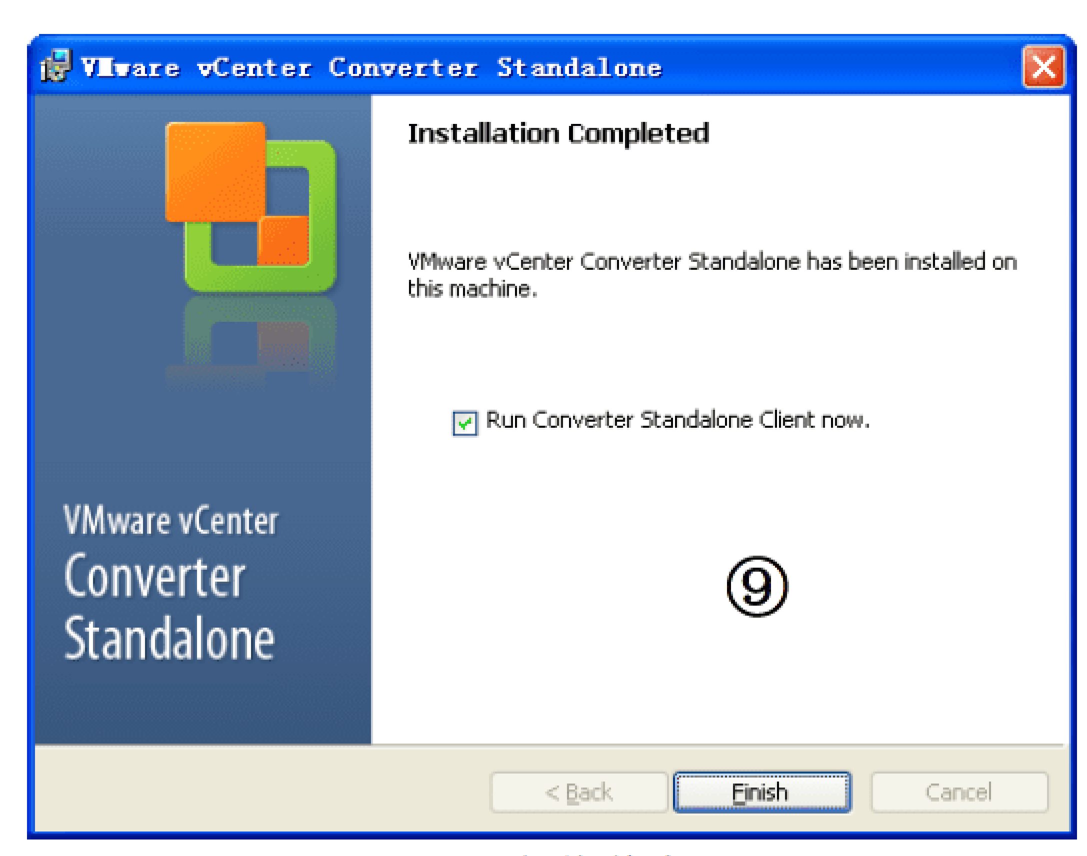
8. 接下来会开始安装。



▲ 这是准备安装的画面

▲ 开始安装了

9. 当弹出如下图所示的画面时，意味着安装落实。



▲ 安装落实

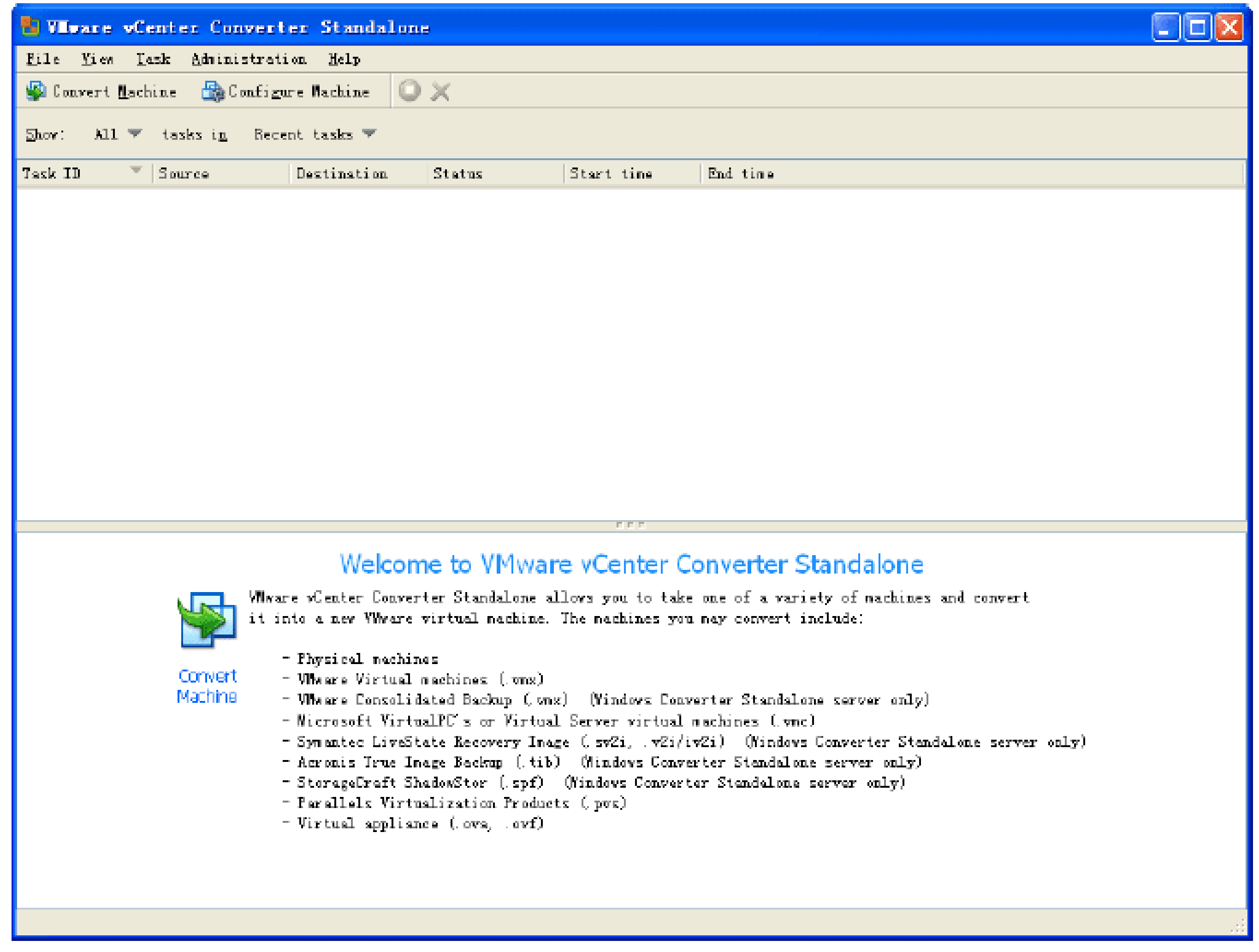
2. 使用 VMware Converter 进行 P2V

在安装好了 VMware Converter 之后，我们就可以使用它来进行 P2V 了。使用 P2V 时，和

VMware Workstation 一样，VMware Converter 会在物理机上安装 VMware Converter Agent 程序，因此必须将物理机的网络联机开放。下面就是转换的步骤。

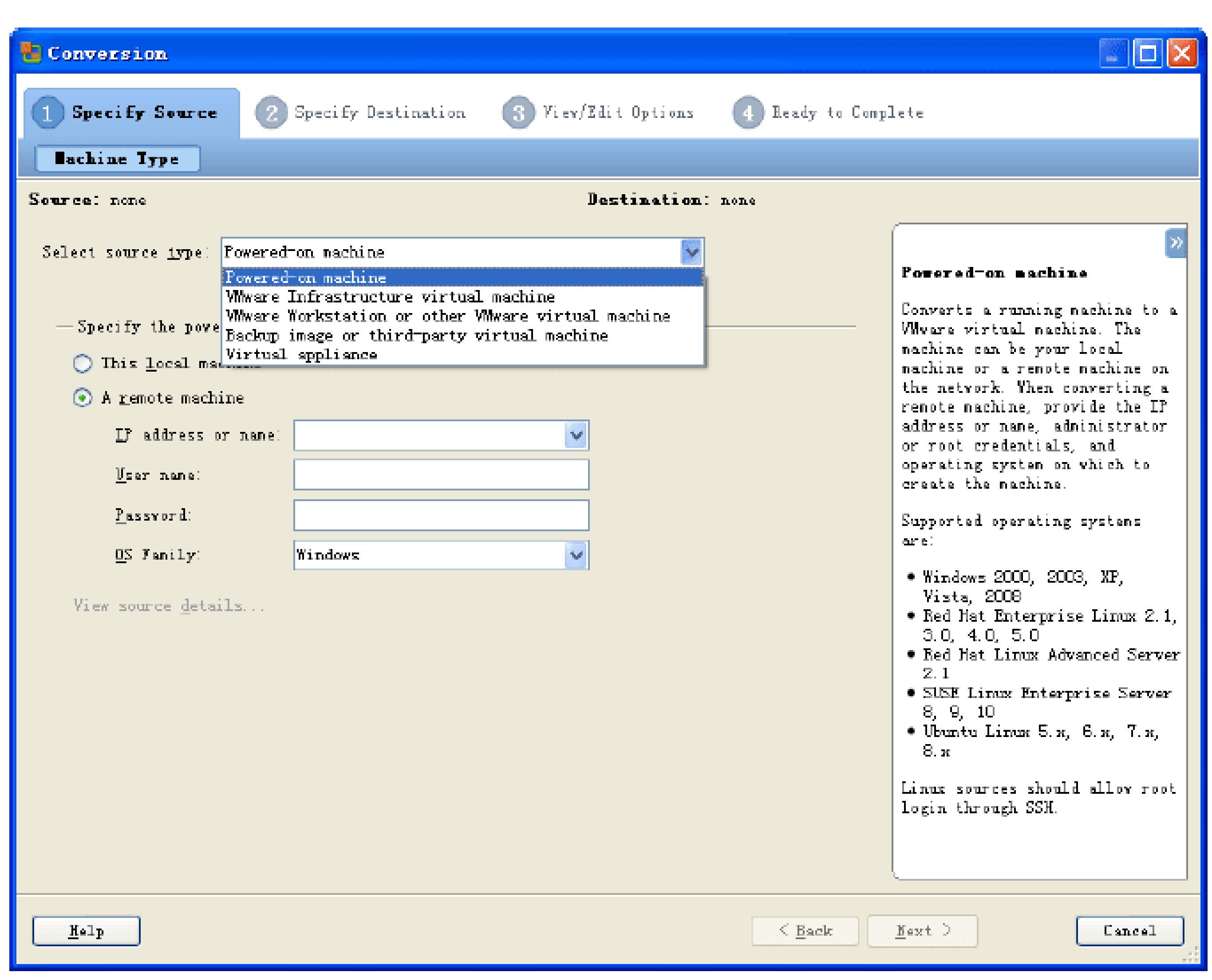
▶ 使用 VMware Converter 来进行 P2V

1. 运行 VMware Converter，选择 Convert Machine 选项。



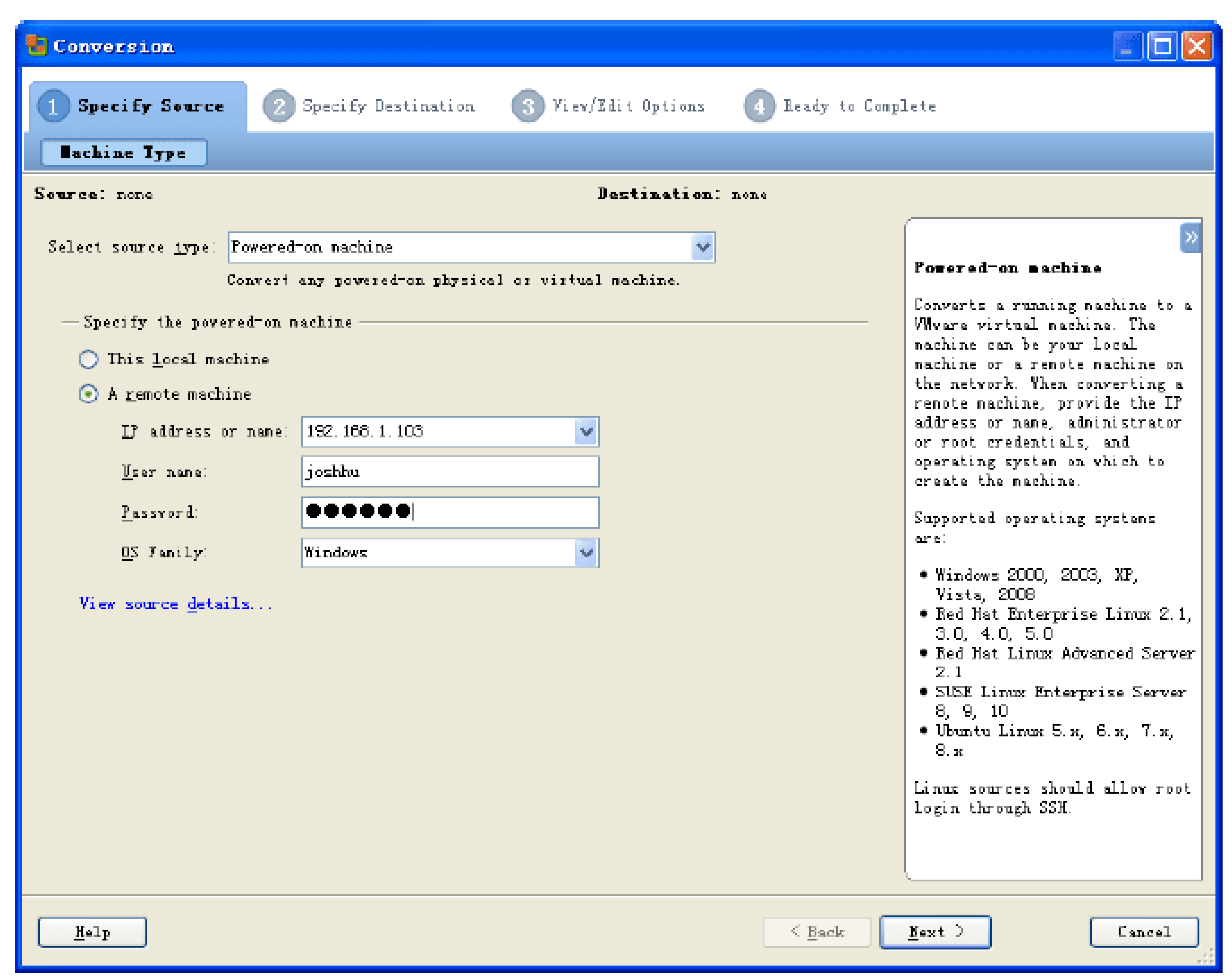
▲ 选择选项

2. 接下来可以选择转换的类型。VMware Converter 支持更多的类型，包括了引导时的物理机，vSphere/VI 的虚拟机，其他 VMware 的产品（如 VMware Workstation）、VA 格式的虚拟机或是副本的图像。我们就选择引导的主机。



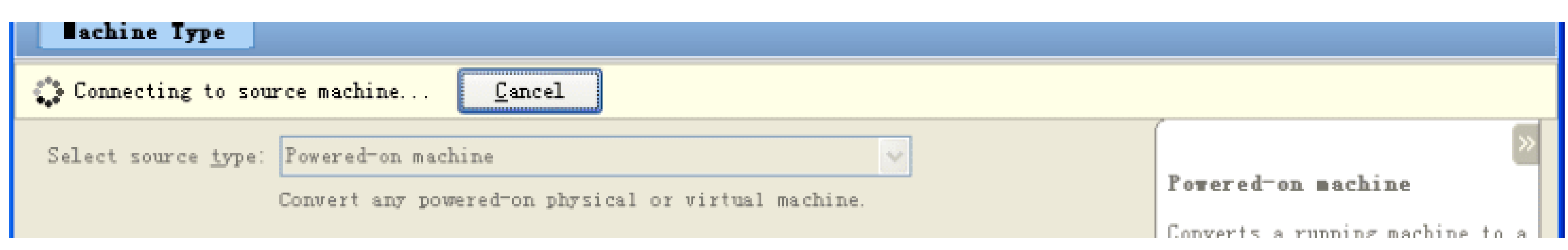
▲ 第一项是运行中的主机

3. 我们选择引导的物理机之后，键入该主机的 IP，系统管理员的账号口令，并且选择操作系统的家族，单击 Next 按钮继续。



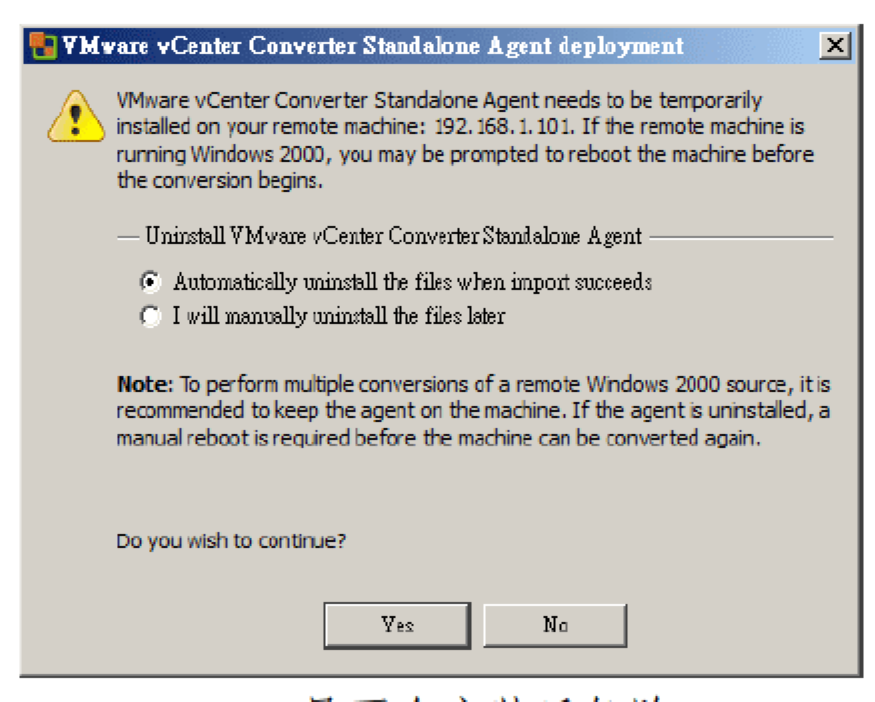
▲ 键入该主机的账号、口令及 IP

4. 接下来会进入连接至目标机器的动作。



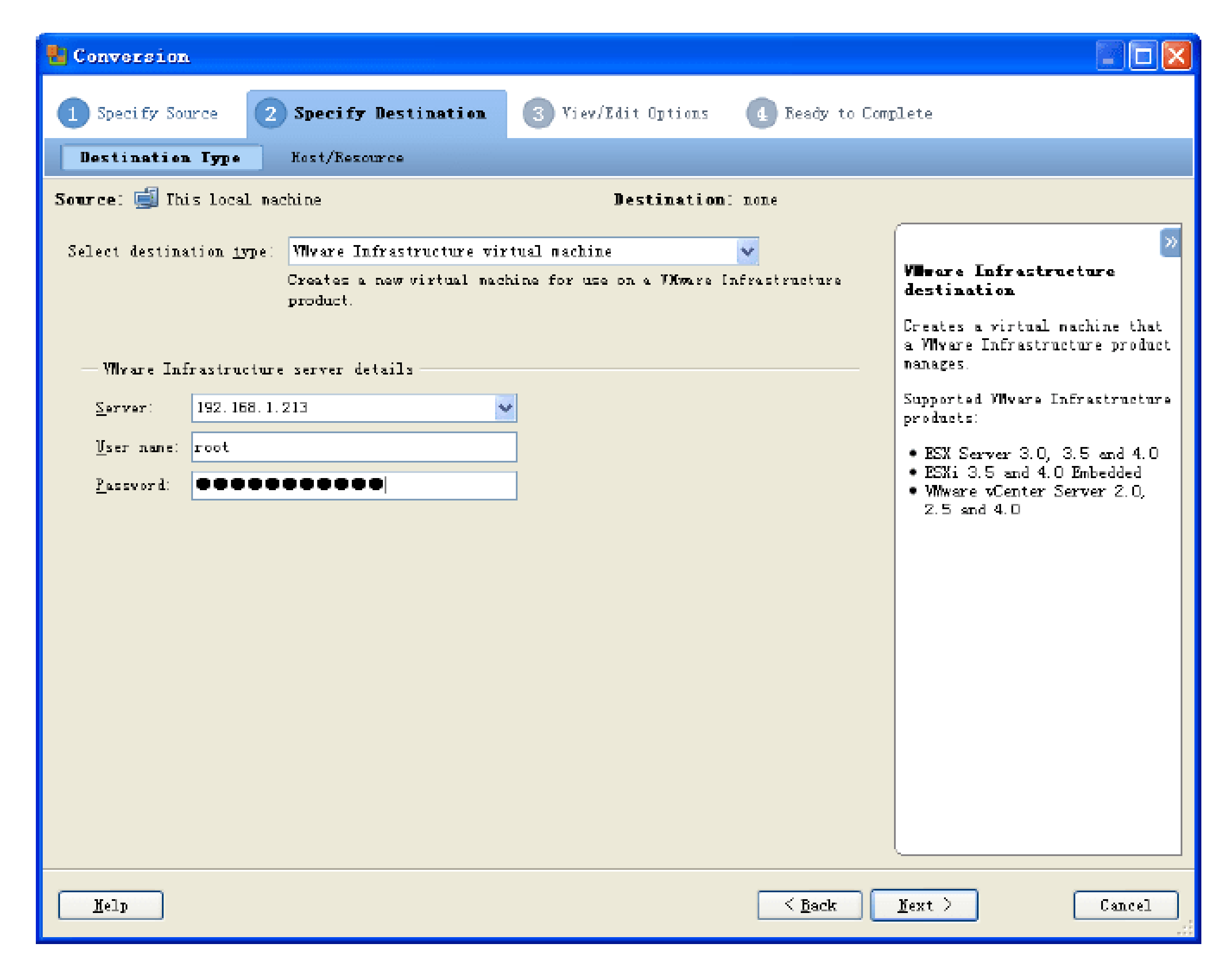
▲ 系统会试着将 Agent 安装到物理机上

5. 当 Ping 通之后，也会和 VMware Workstation 一样询问是否要在安装后卸除任务，选择第一项单击 Next 按钮继续。



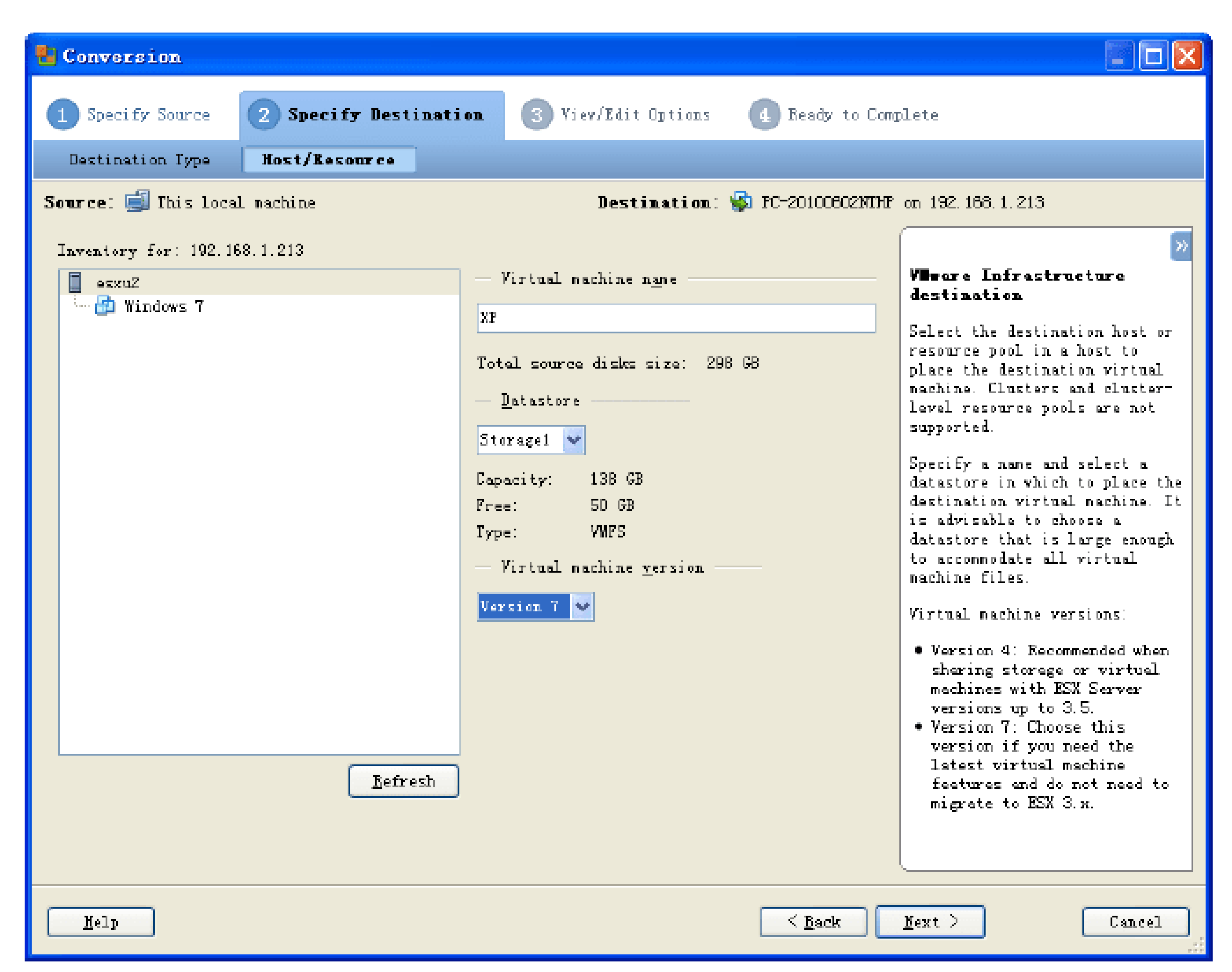
▲ 是否在安装后卸除

6. 接下来会询问要转换的目的地。我们选择 vSphere 的主机，并且键入该主机的账号口令及 IP。单击 Next 按钮继续。



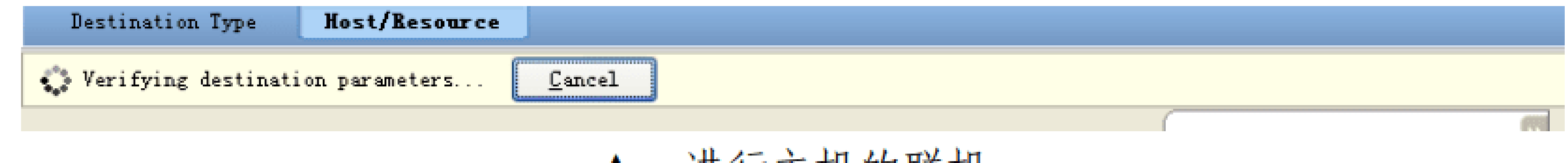
▲ 选择 ESX 主机的账号、口令及 IP

7. 接下来是替这个 VM 取一个名字，并且选择 Datastore 及 VM 的版本。单击 Next 按钮继续。



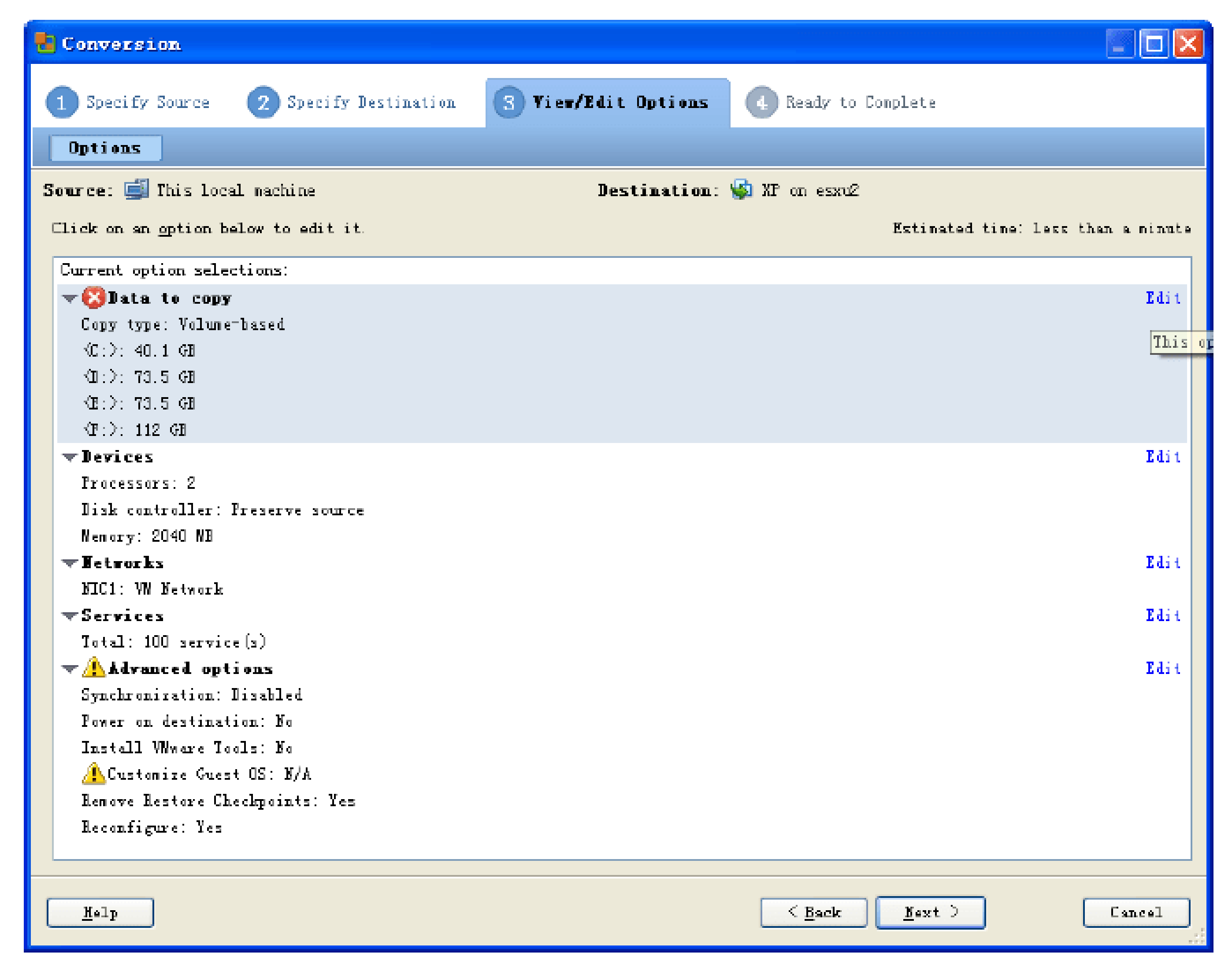
▲ 替 VM 取一个名字，并且选择目的及 VM Hypervisor 的版本

8. 接下来则会将在图中连接至 ESX 主机上。



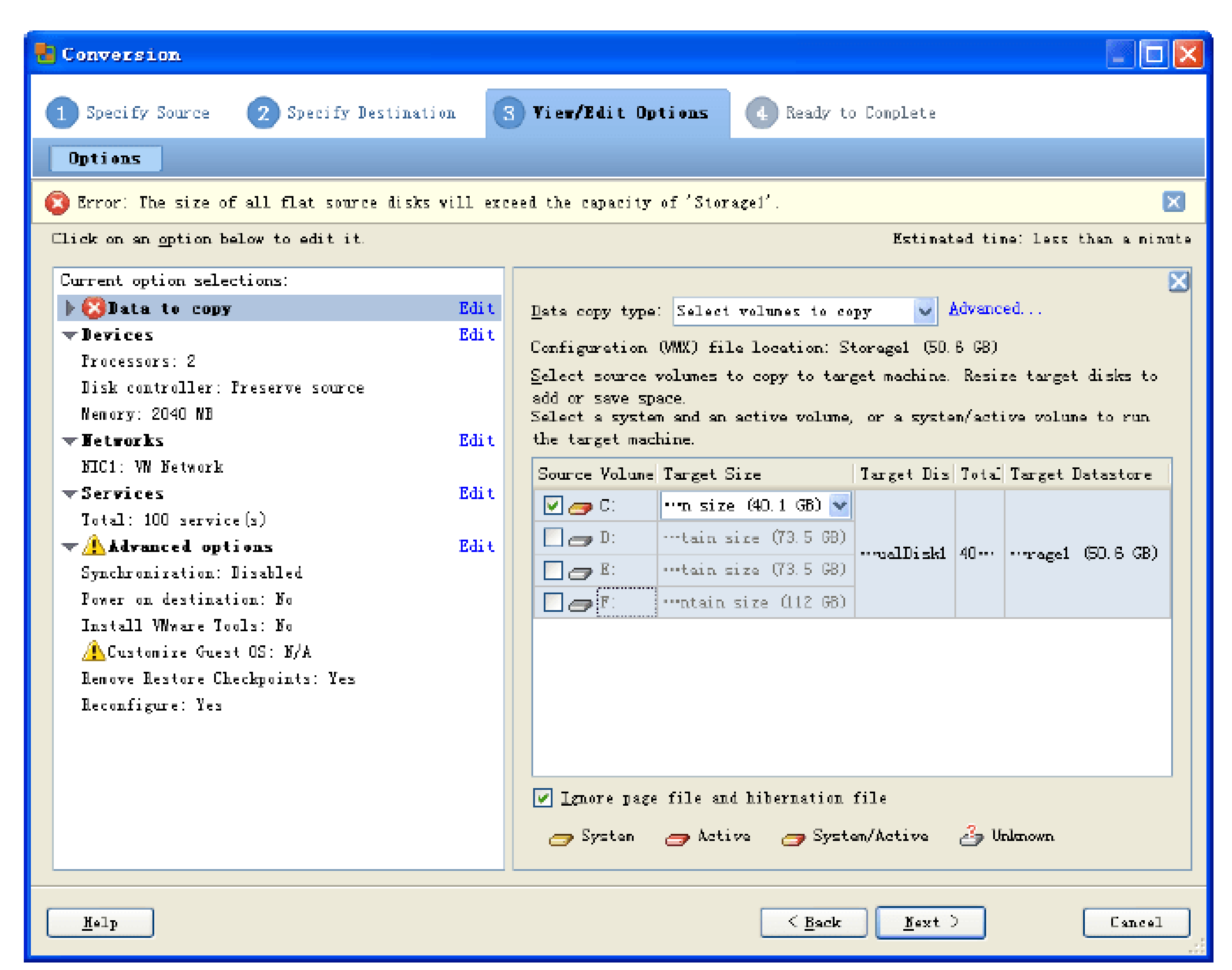
▲ 进行主机的联机

9. 连接完毕之后，会给出该来源机的一些配置。我们必须在线更改。在图中我们可以看到有打叉的地方，就是无法转换的部分，我们可以单击右边的 Edit 进入修改。



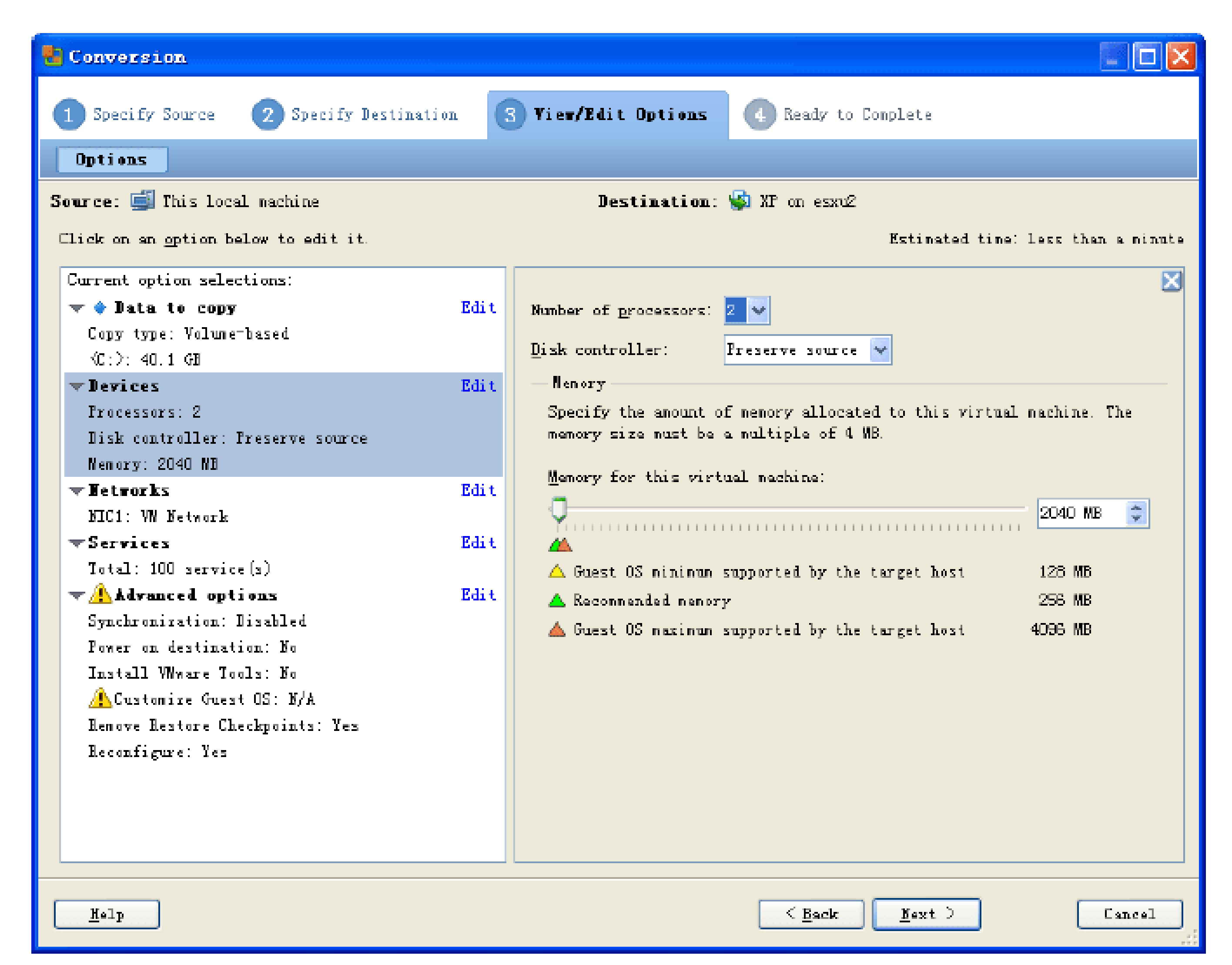
▲ 这里是需要修改的

10. 举例来说，我们只要转换 C 磁盘，就可以把其他的磁盘不打勾。



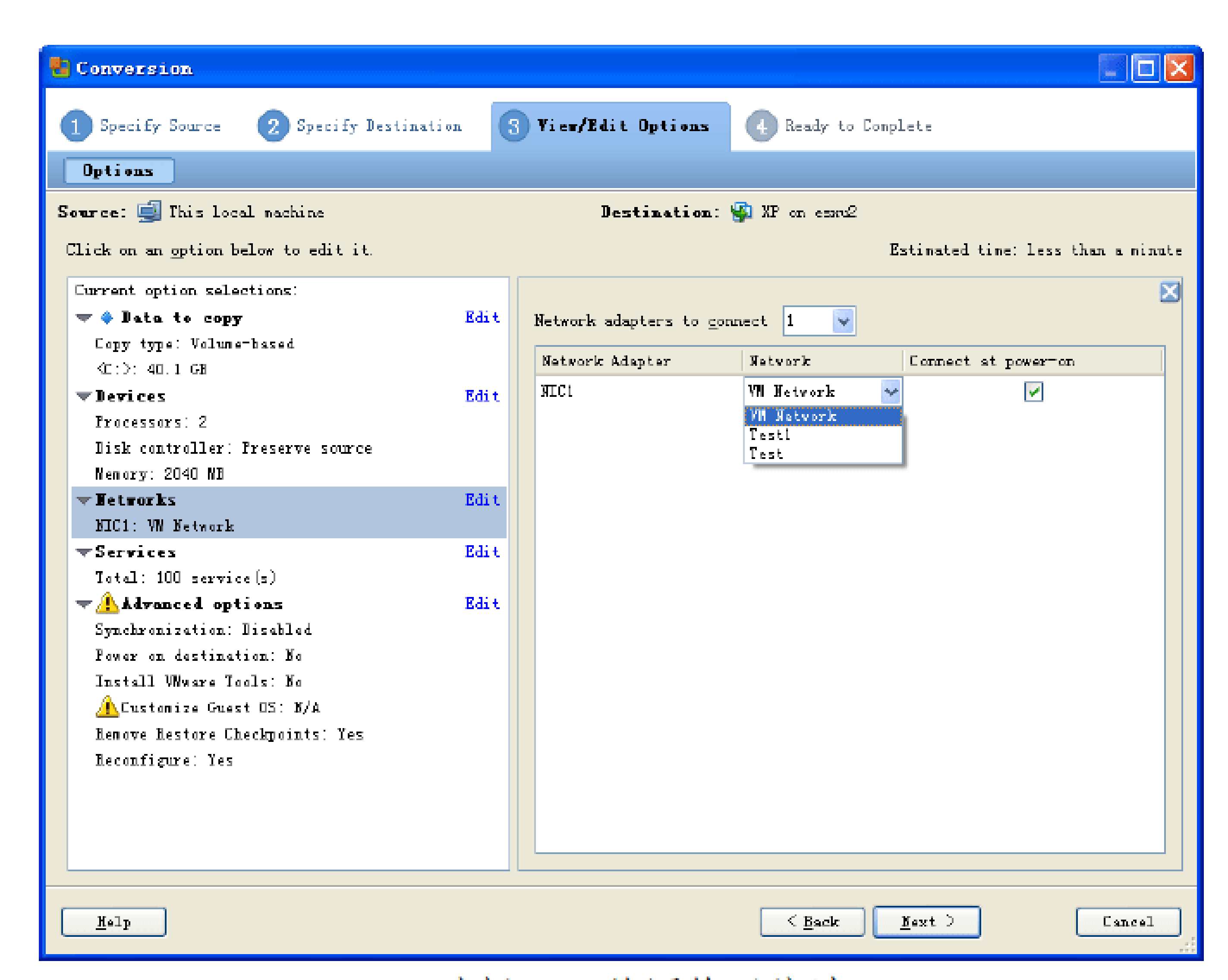
▲ 只转换 C 磁盘

11. 也可以在线调整硬设备，如 CPU 数量及内存大小。



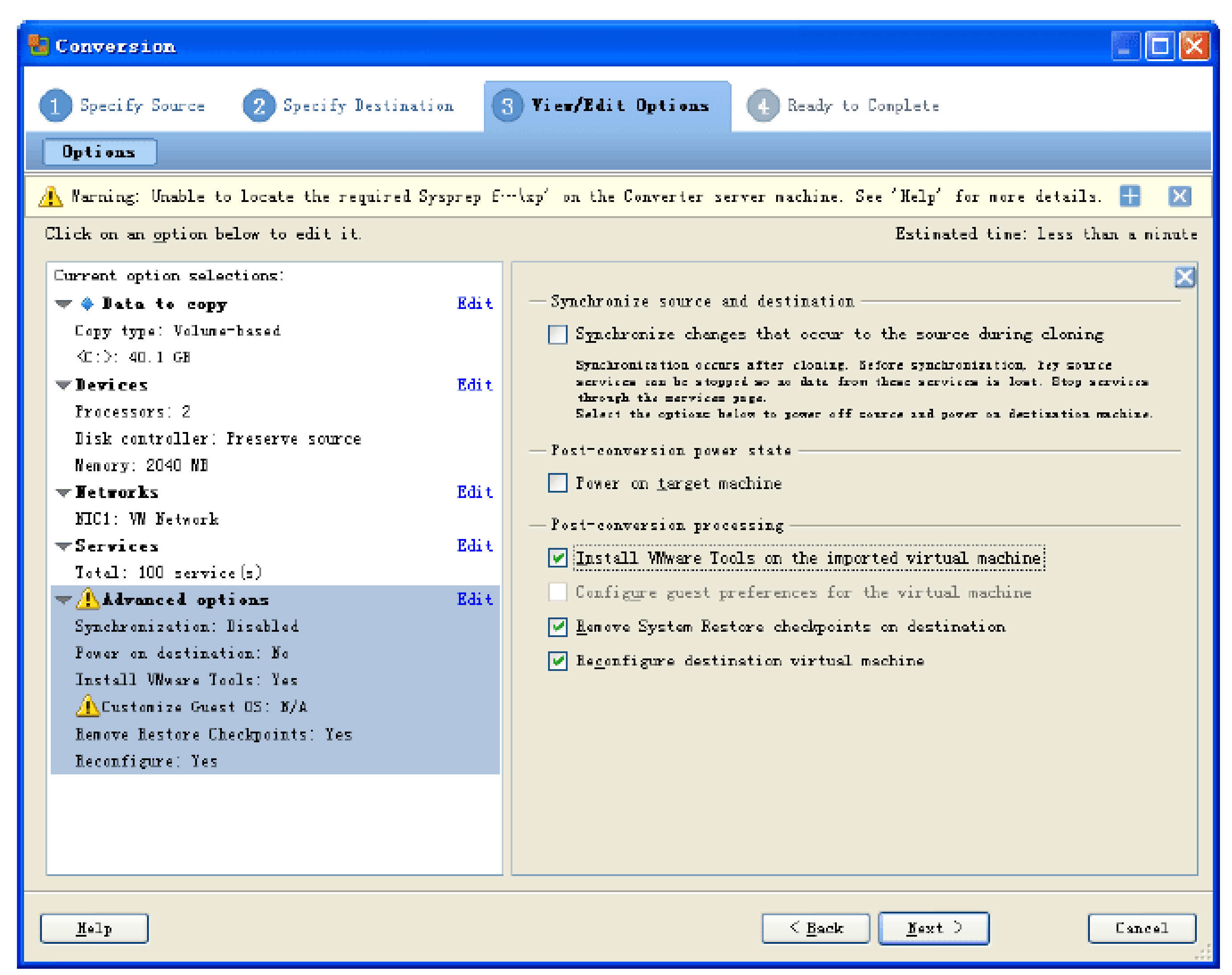
▲ 在线调整 VM 的硬设备

12. 别忘了到 Network 的部分选择 VM 的网络通信端口。



▲ 选择 VM 的网络通信端口

13. 在 Advanced options 中，别忘了选择一些参数。其中可以安装 VMware Tools。如果不是作范本机，那最后一项 Reconfigure destination virtual machine 别忘了撤销。



▲　这台也不做模板机，因此不选择配置细部选项

14. 我们也可以进入选择物理机和虚拟机引导时的选项。

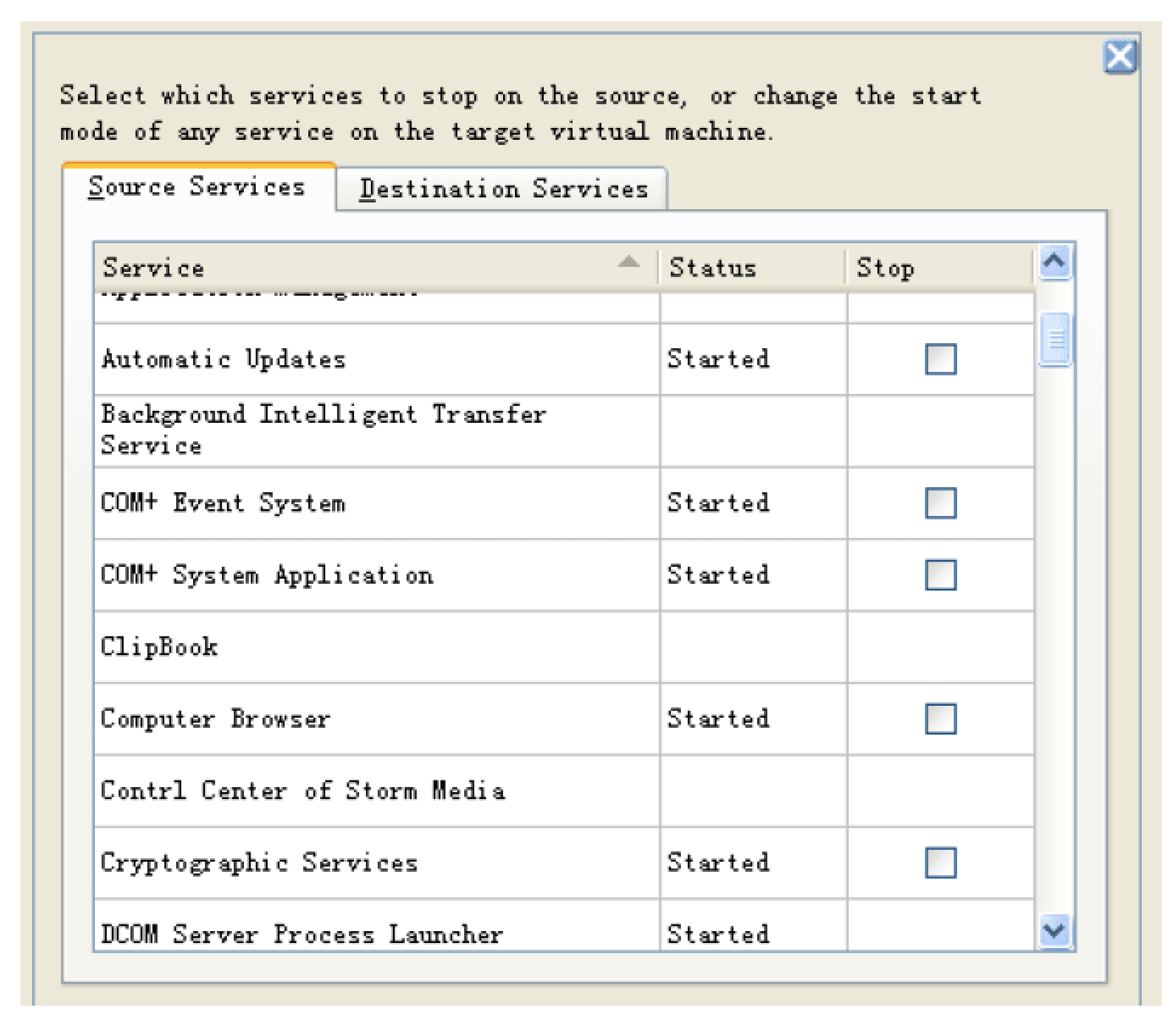
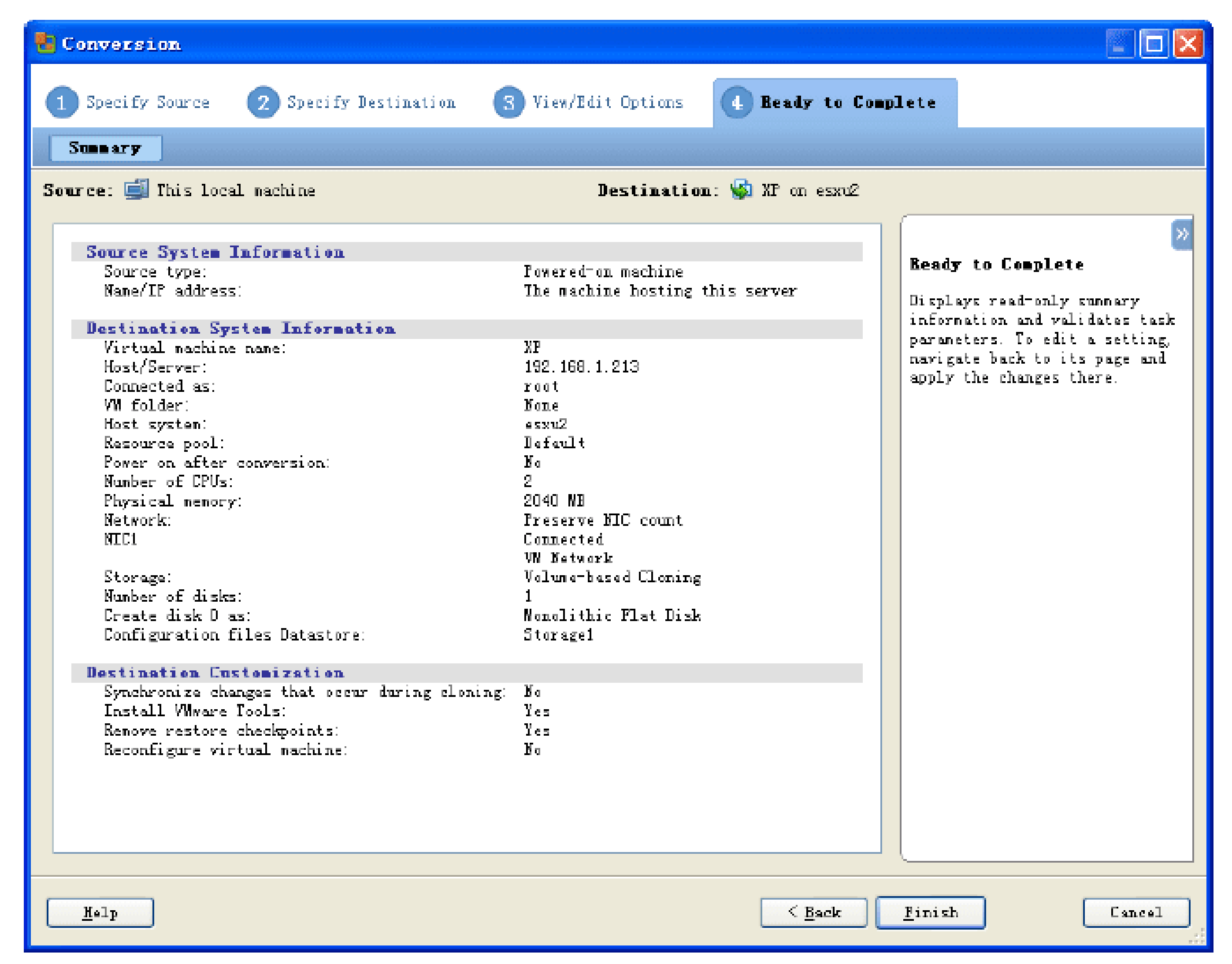


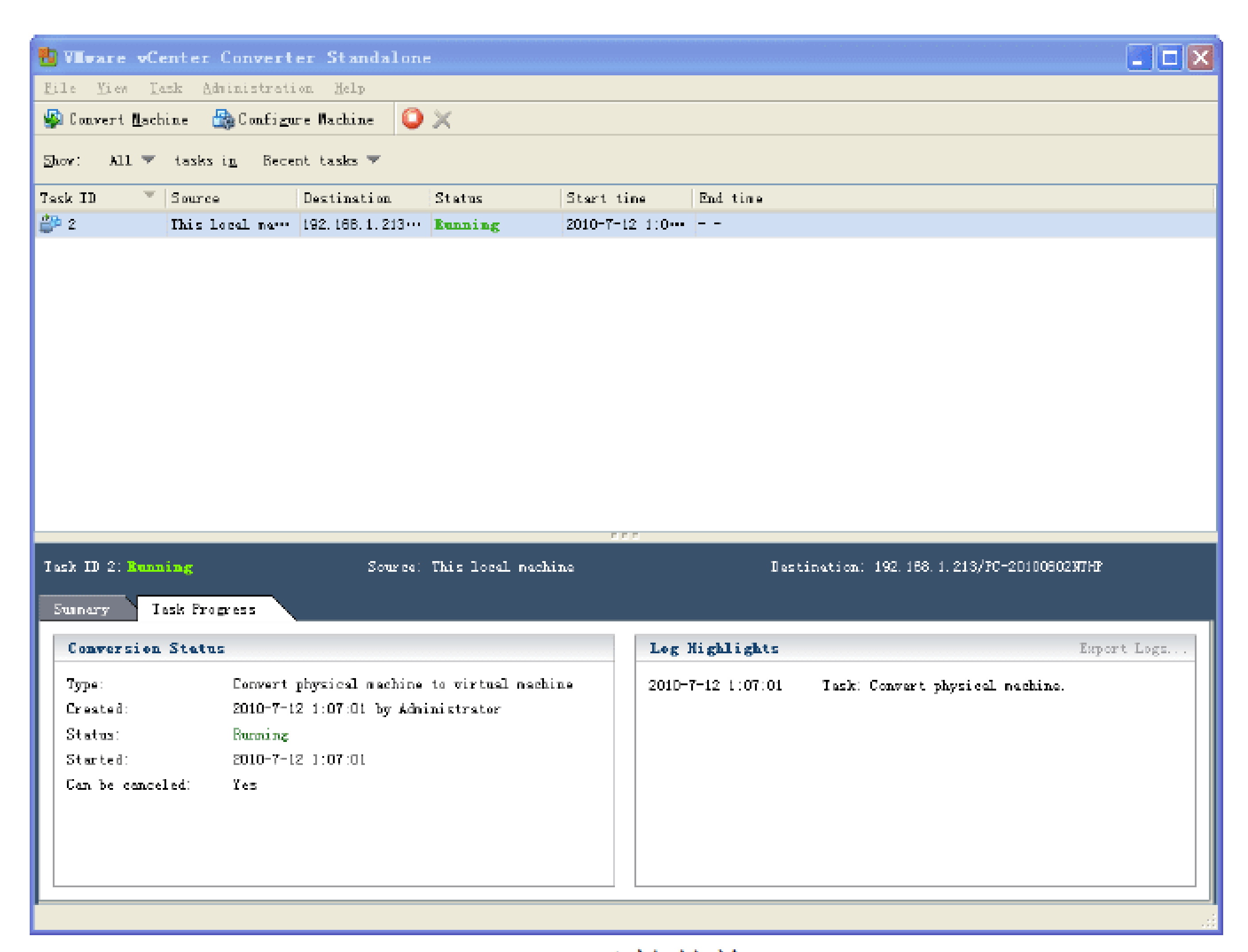
图 24-56　这里是物理机或 VM 引导时的服务选项

15. 当配置完毕之后，进入总结画面，单击 Finish 按钮就可以进入转换了。



▲ 总结画面

16. 此时会进入 VMware Converter 的主画面，开始进行转换。通常转换会花上数十分钟到数小时，看你的物理机的硬盘大小而定。



▲ 开始转换

24.3 使用第三方工具进行 P2V

使用 VMware 提供的工具来转换是最方便的。但有时使用 VMware 的工具也会失败，由于 VMware Converter 必须在来源机上安装 Agent，但如果网络的配置有问题，就会导致安装的失败。我们这时可以将 VMware Converter 安装在要转换的机器上，使用“直接转换本地”的功能，但这会让物理机的服务被影响，因此我们也可以借助第三方的工具，本章就来看看两个较常用的工具，一个是 True Image 的副本，另一个则是小巧的 WinImage。

24.3.1　使用 Acronic True Image Echo Server

对想要虚拟化的企业来说，最重要的一个步骤就是将现有的机器转换成虚拟机。MIS 人员常常等待所有的同事都下班之后，才能将机器进行转换。但在网络发达时代，机器是没有“下班时间”的。也许同事下班了，但他们还是会在家里收信，或是工程师在跑一个报表总汇，需要用到 SQL 一整夜，这时候你也无法进行转换了。另外也有很多人曾经问我能不能将 Ghost 的图像文件转换成 VMDK，事实上，这两个是同一件事。想在线将当前使用的物理机副本或是转换成虚拟机，都是 MIS 环境常见的，这里推荐一个好工具，就是 Acronis True Image。

Acronis True Image 主要分为服务器上的版本和 Windows XP/Vista/7 下面的 Home 版。除了多台机器进行副本还原和管理的功能之外，Home 版的功能已经够强了（针对单个机器而言），但如果你要使用 P2V 功能的话，还非 Server 不行。我们在这一小节中，就来看看用 Acronis True Image Echo Server 来将副本文件转换成 VMware 的 VMDK。

▶ **使用 True Image P2V 的步骤**

（1）安装 Acronis True Image Echo Server。

（2）副本系统分区。

（3）转换副本档至虚拟磁盘。

（4）创建 VMware 的虚拟机挂上虚拟磁盘。

（5）引导安装 VMware Tools。

1.　安装 True Image

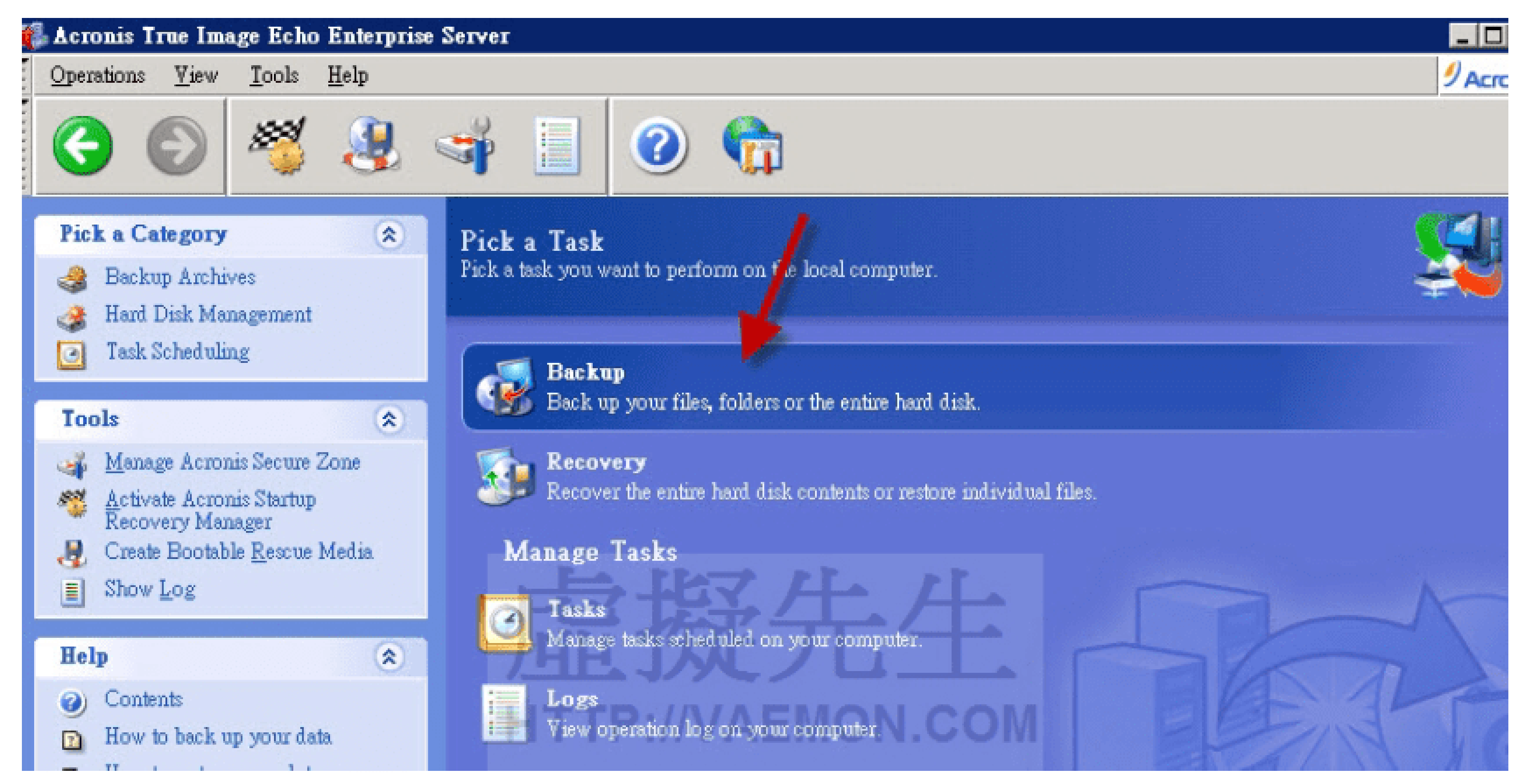
True Image Echo Server 是一个付费软件，而且只能安装在 Windows 2003/2008 服务器环境，最好是安装在有 AD 集成的环境中，才能对域中的计算机进行远程副本。当然安装这个软件就像任何一个 Windows 软件一样，只要一直单击下一步即可，读者们只要在购买时直接安装起来即可。

2．使用 True Image 副本要转换的分割区

在转换之前，我们必须先将分割区副本成 True Image 的格式。因此整个步骤是先副本分割区再进行转换，这里先看看副本的步骤。（作者注：本章图片使用作者网站上图片，因此会有水印）

▶ **使用 True Image 创建副本**

1. 进入桌面上的 True Image Echo Server 的图示。
2. 进入主画面之后，直接选择 Backup 链接。



▲ 选择 Backup 链接

3. 此时会进入 Backup 的欢迎画面，单击 Next 按钮继续。



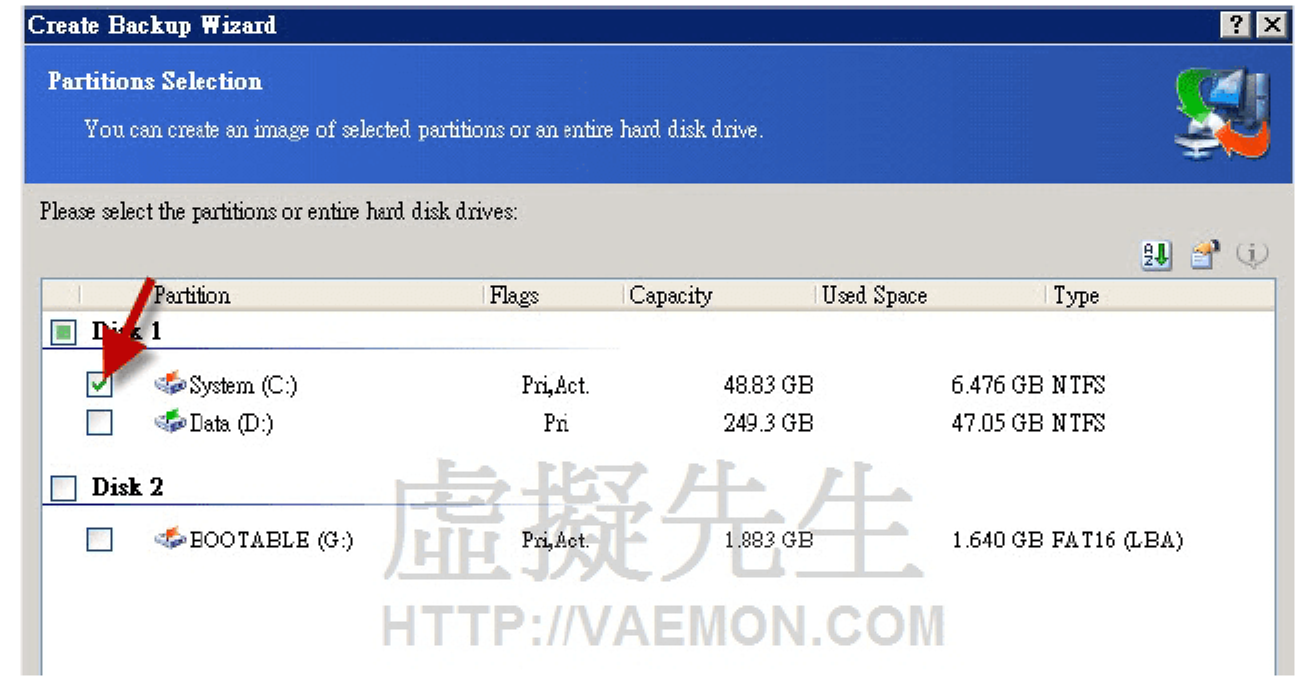
▲ 欢迎画面

4. 在 Home 版中，会让你选择“分区”，但这里是用“计算机”和“数据”来分别的，因此我们要副本整个系统分割区，就得选择 My Computer，单击 Next 按钮。



▲ 副本计算机而非数据

5. 选择要副本的系统分区。True Image 会将这台计算机中所有“物理硬盘”的“第一个分区”预先起来，包括 USB 硬盘。就选择你要的系统分区即可。单击 Next 按钮继续。



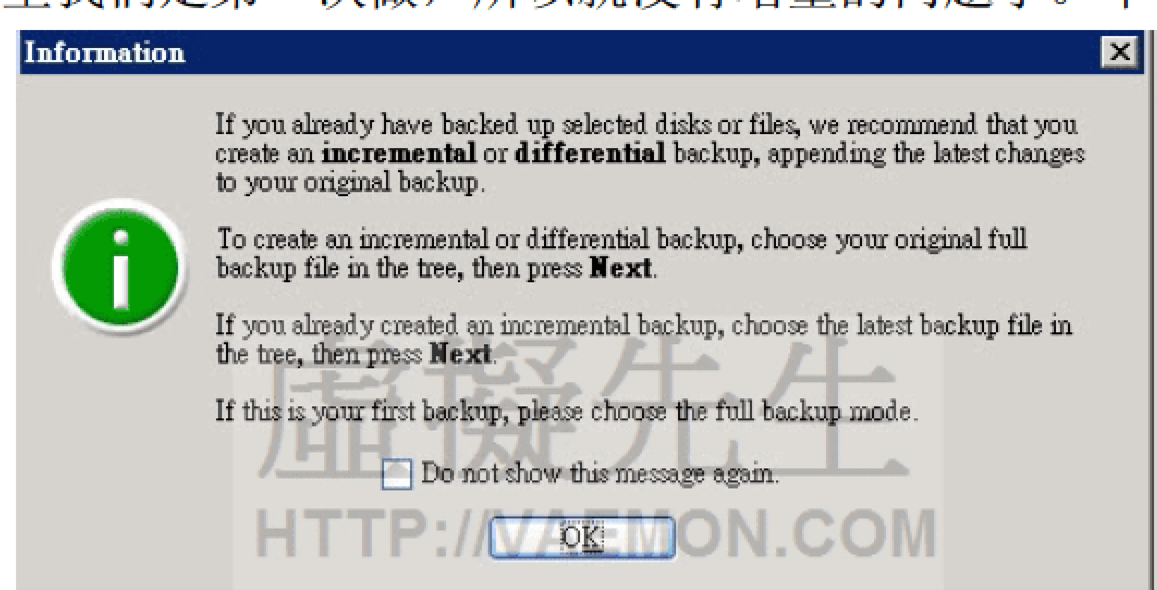
▲ 选择分割区，通常 P2V 只会副本系统分割区

6. 接下来是排除选项，一般来说可以都不选，前提是你的系统将实用程序、数据都分层好了。这里就有“排除 SQL 文件”的选项。直接单击 Next 按钮。



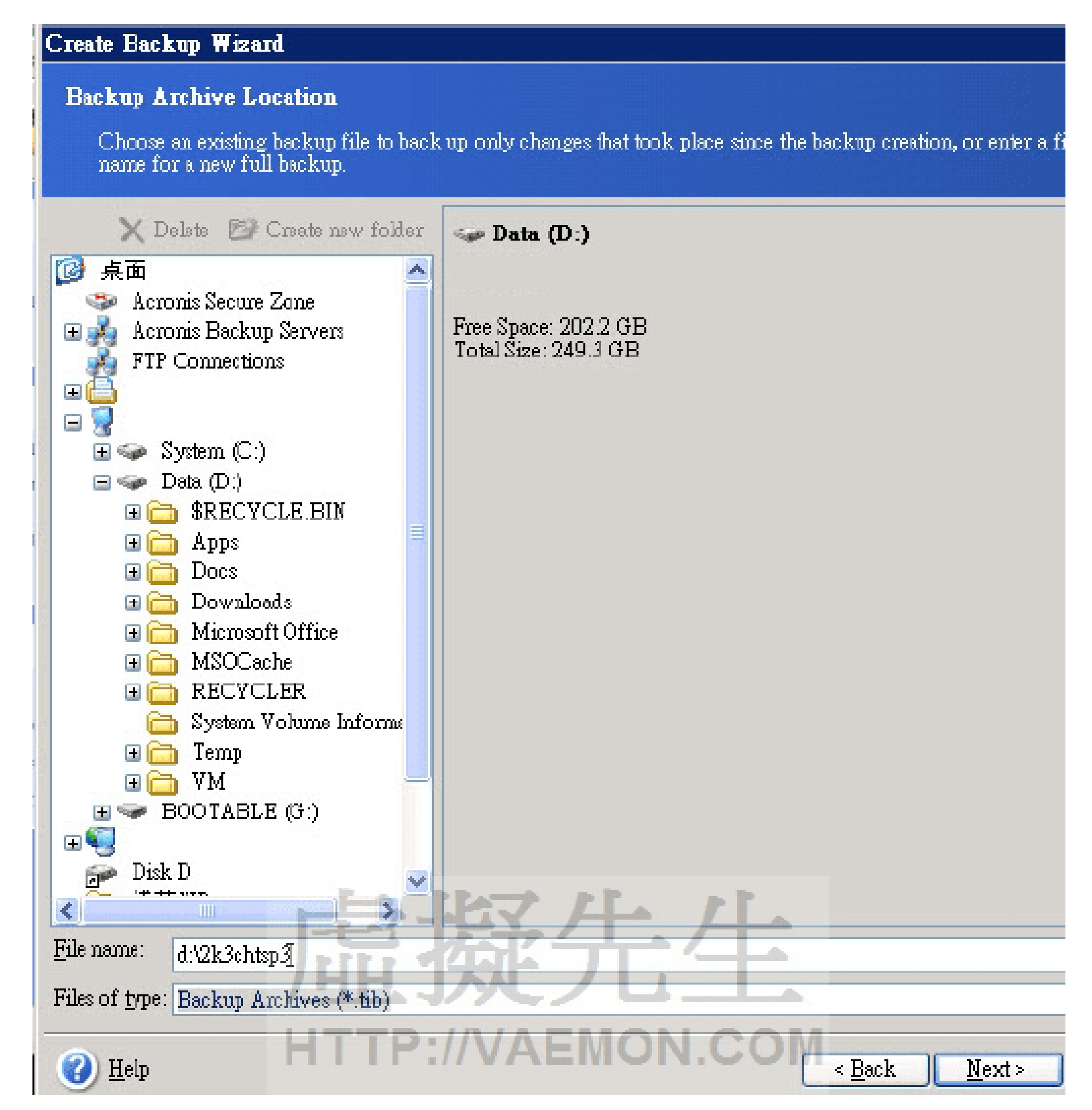
▲ 在副本时，可以选择不要副本 Pagefile

7. True Image 可以进行增量副本，你可以在安装好机器时做一次，然后所有基本实用程序安装好再做一次增量，这样就有点像虚拟机的“快照”功能。可以随时还原到某一个时间点。这里我们是第一次做，所以就没有增量的问题了。单击 Next 按钮继续。



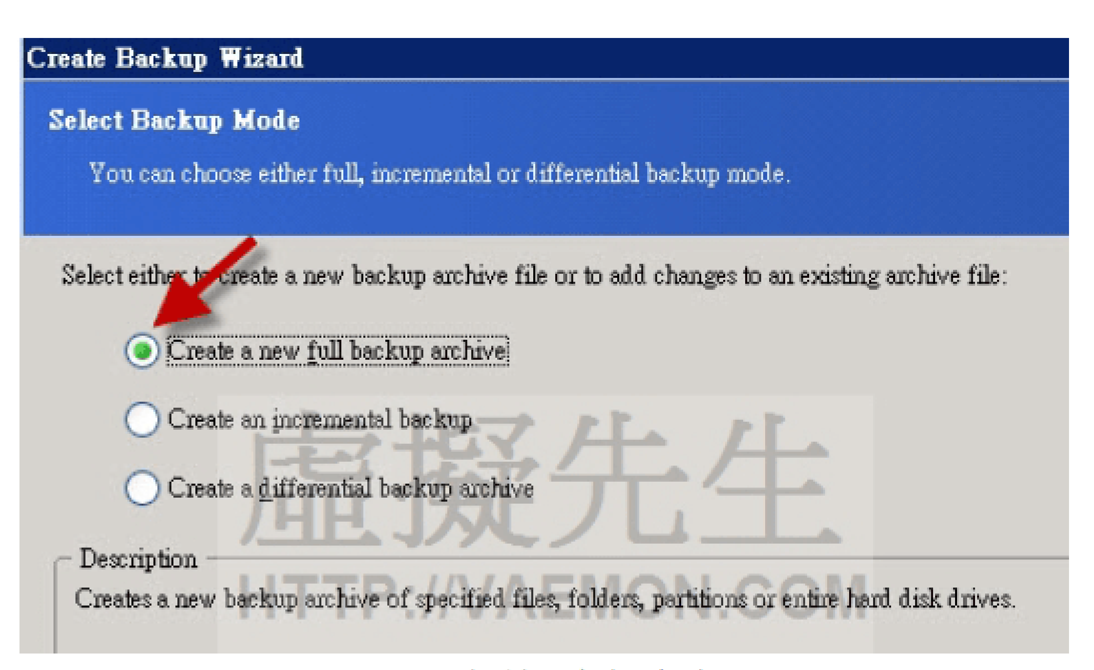
▲ 选择全副本就行

8. 键入副本的地点以及文件名。当然不是能副本的那个分割区，通常会放在第二颗硬盘或网络硬盘上。单击 Next 按钮继续。



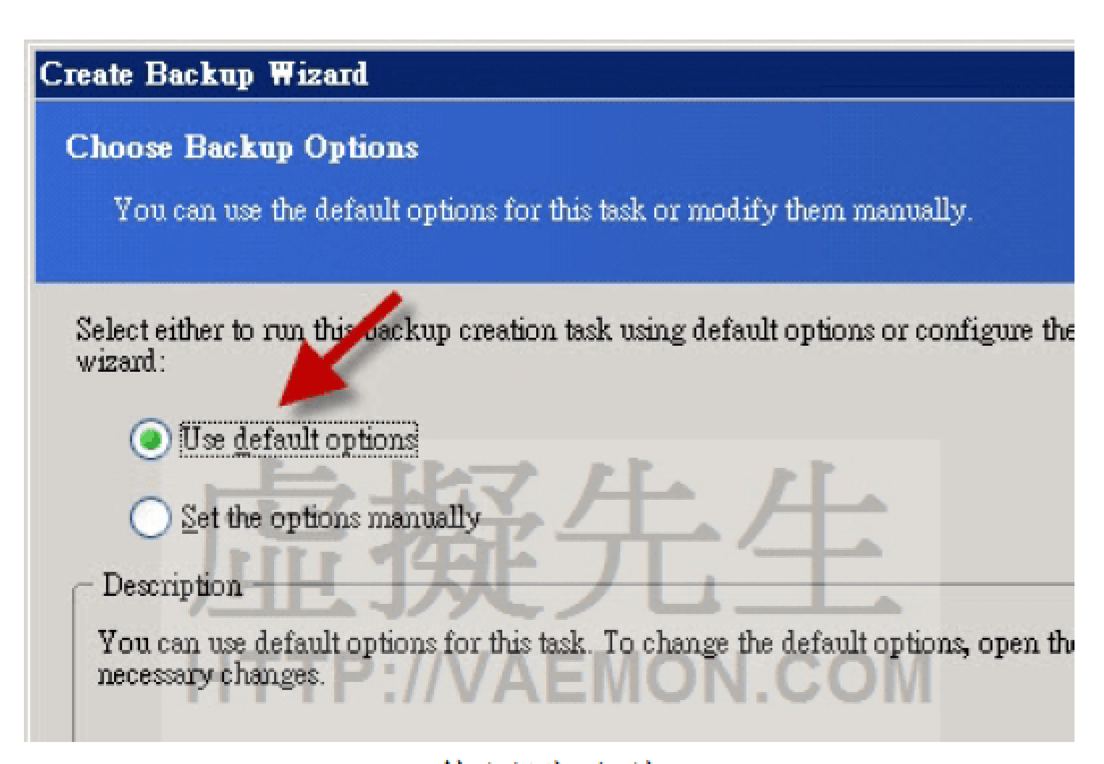
▲ 选择一个适合的目的地

9. 这里就是前面提供的全副本、增量副本和差异副本。选择“全副本”。单击 Next 按钮继续。



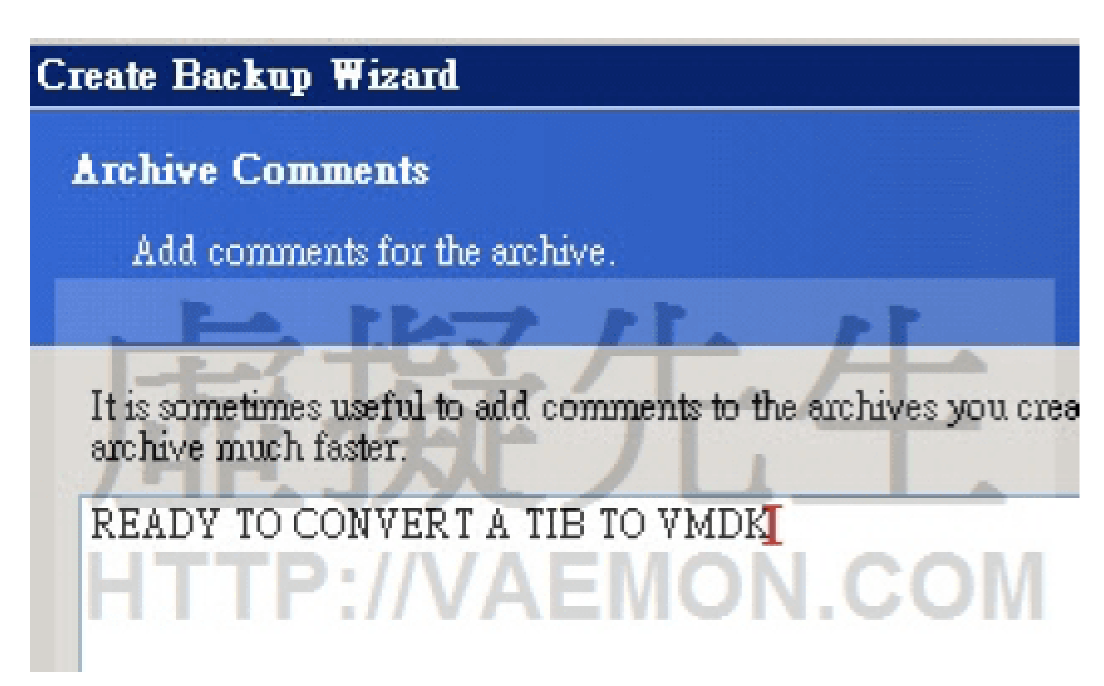
▲ 直接副本全部

10. 接下来是配置副本选项，包括账号口令、加密方式、是否要可写批注。如果在企业环境，可以选择“配置选项”，但如果你只是要转换，就选择 default options 即可。单击 Next 按钮继续。



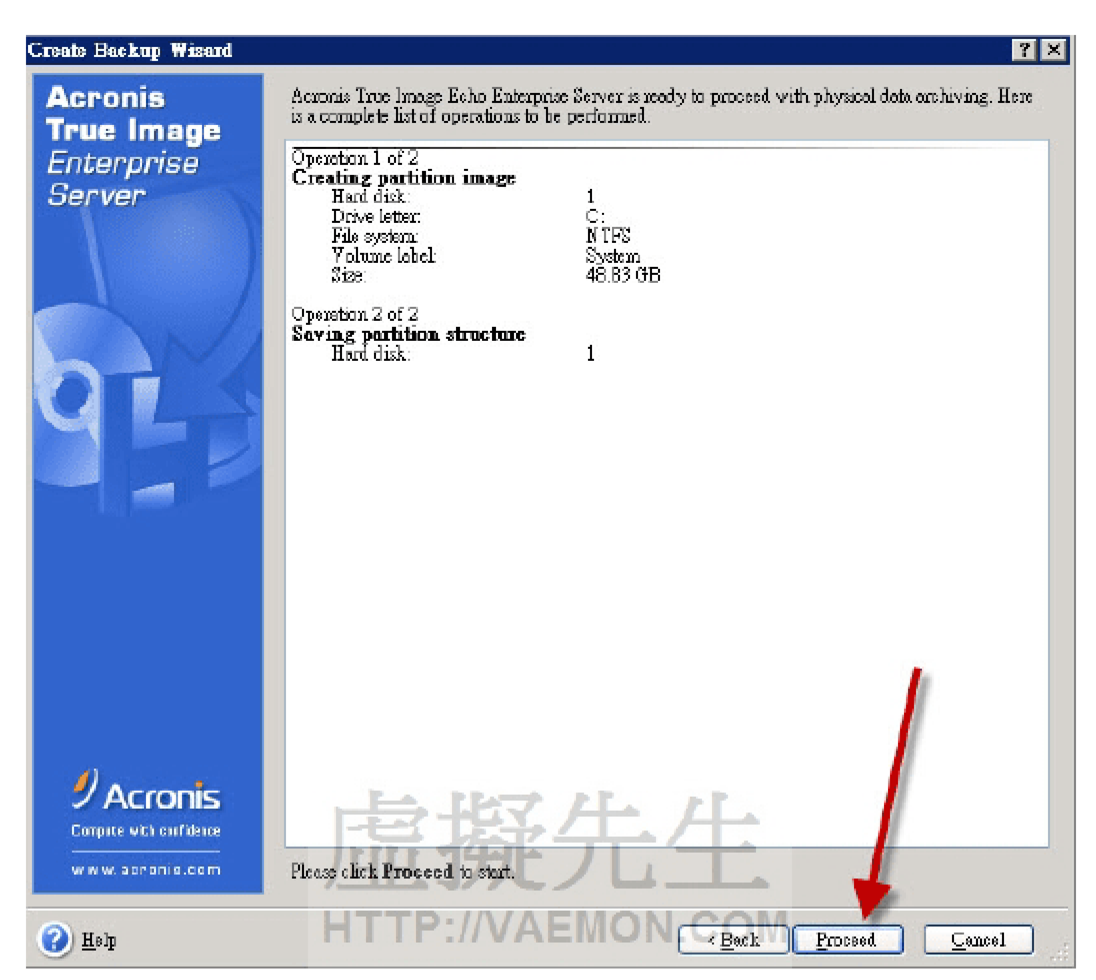
▲ 使用默认值即可

11. 加入批注，可以让你知道每一个副本的用处。单击 Next 按钮继续。



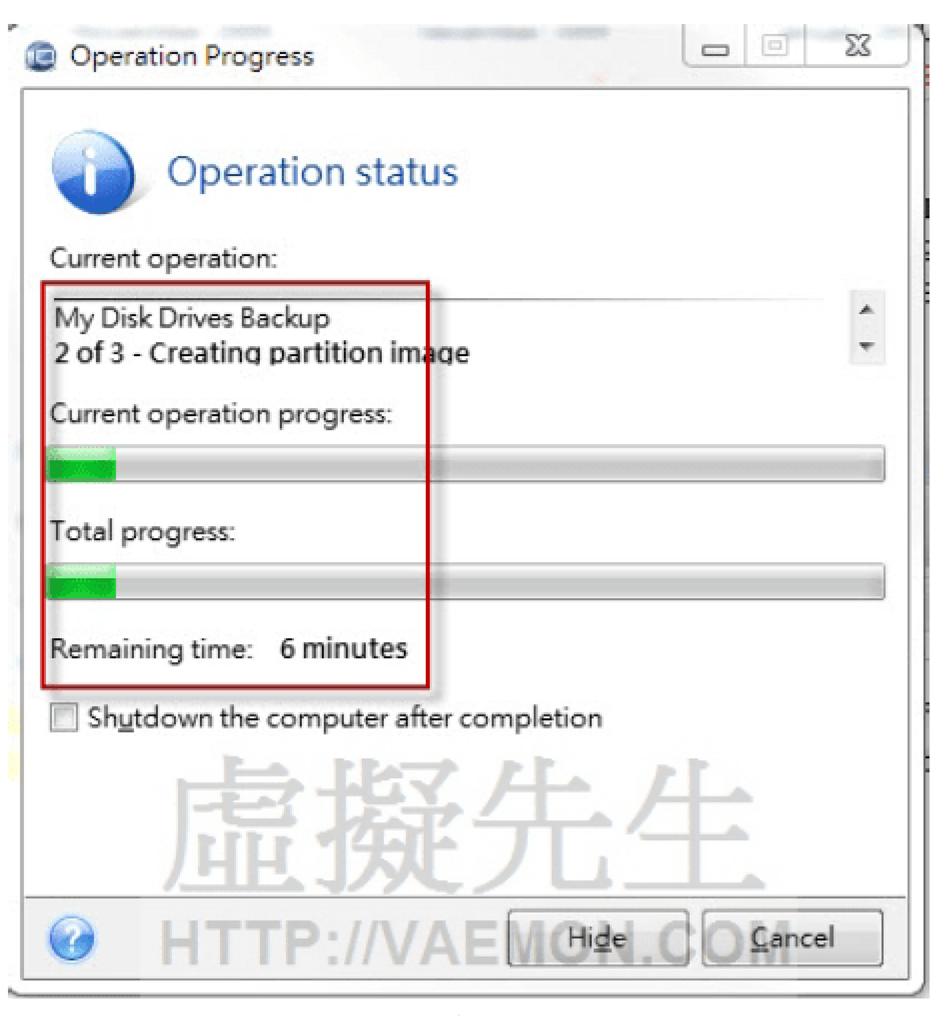
▲ 加入批注

12. 弹出最后确定画面了，直接单击 Proceed 按钮。



▲ 总结画面

13. 开始转换，如果你不想这个画面挡在屏幕中间，可以单击 Hide 按钮。



▲ 开始转换了

14. 当弹出这个画面时，意味着物理机的硬盘已经副本到 True Image 的 tib 文件中了。



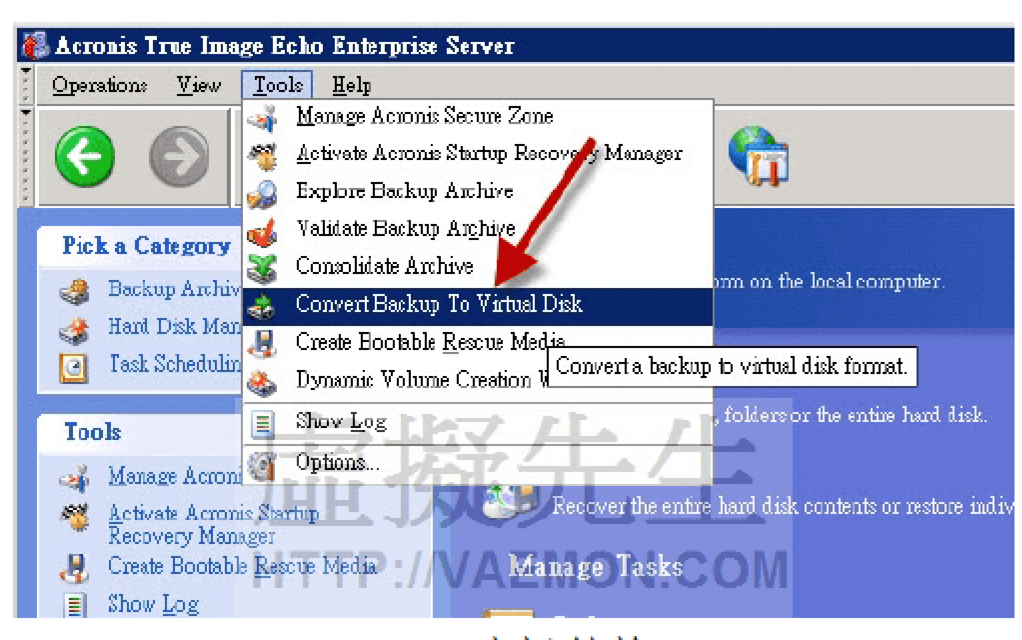
▲ 转换完毕

3. 将 True Image 的副本转换成 VMDK

P2V 的过程就是硬盘的转换，而转换之后的 VMDK 必须能让 vSphere 使用才行。True Image 的转换功能就是这个功效，我们就来看看。

▶ **将 True Image 的副本转换成 VMDK**

1．在 True Image 中选择 Tools 选项卡中的 Convert Backup to Virtual Disk 选项。



▲ 选择转换

2．直接单击 Next 按钮。



▲ 单击 Next 按钮

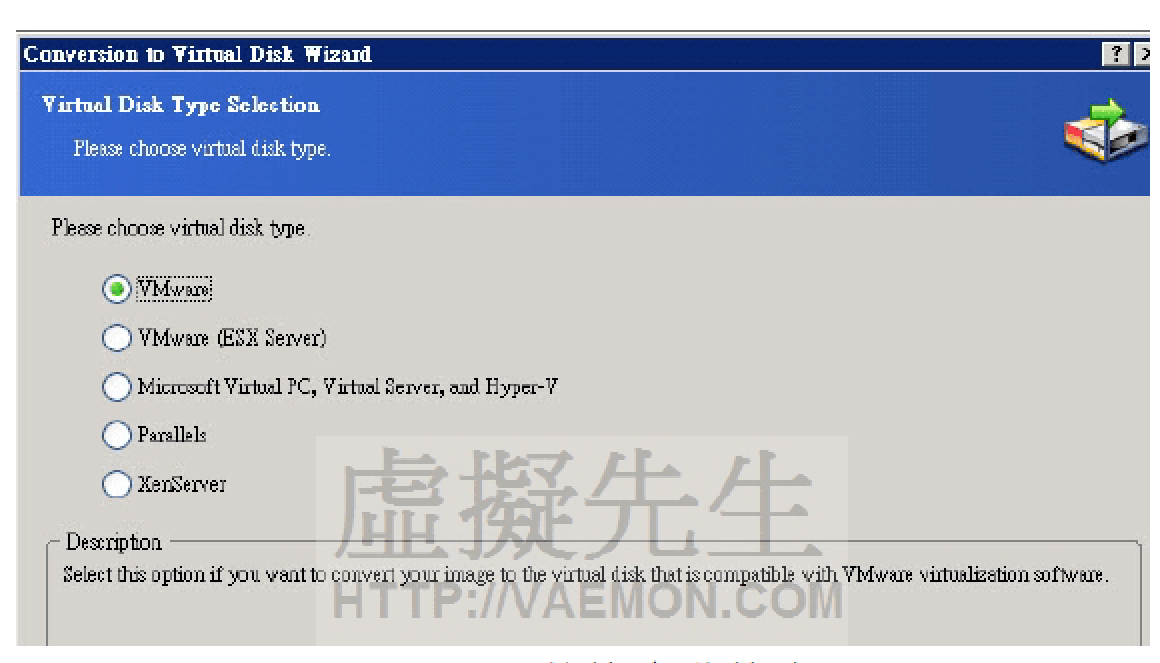
3．选择创建好的 tib。



▲ 选择 tib 格式的文件

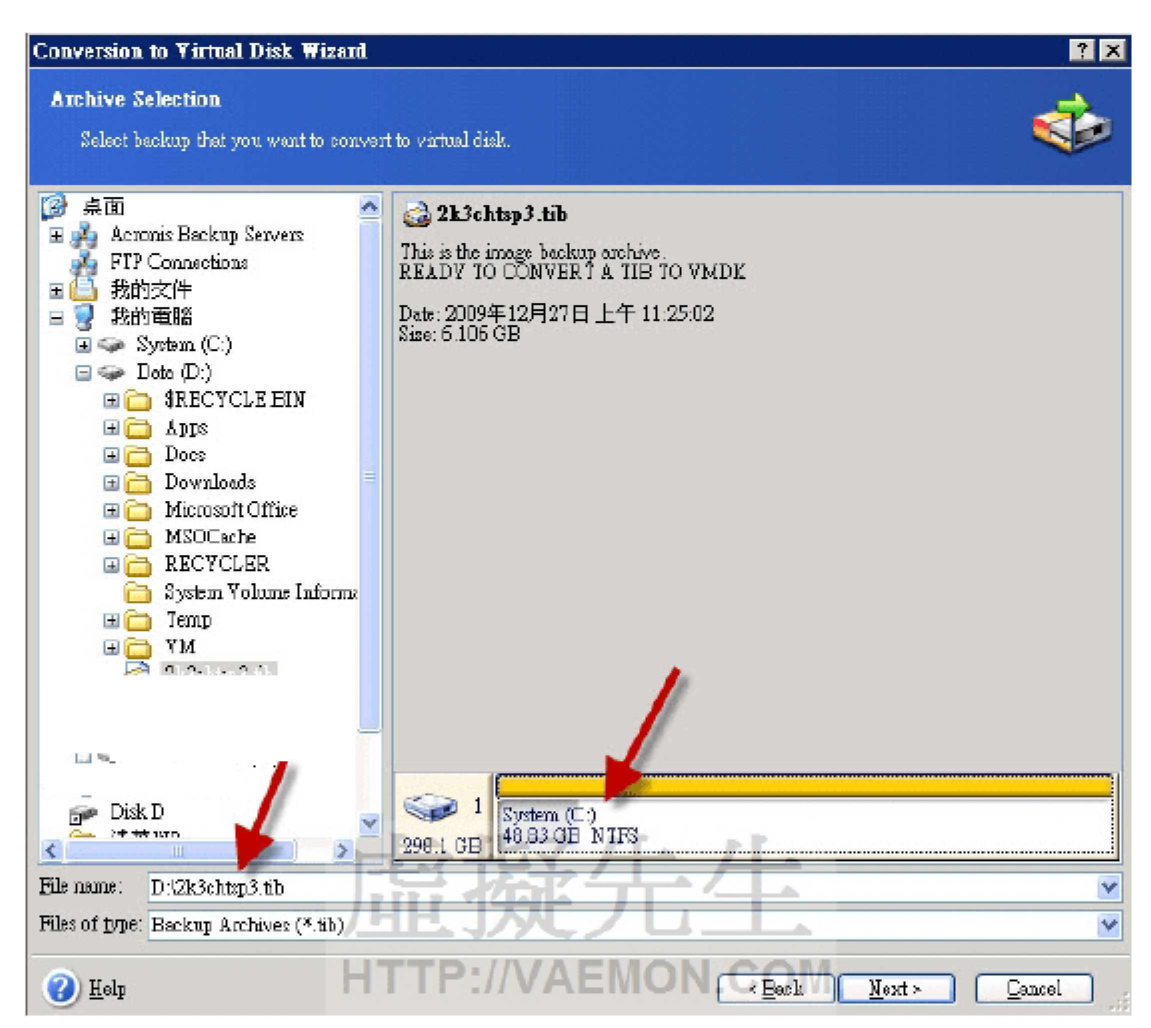
4． 选择格式，最方便的就是这里有许多格式可以选择，注意第一个 VMware 是给

Workstation 用的。单击 Next 按钮继续。



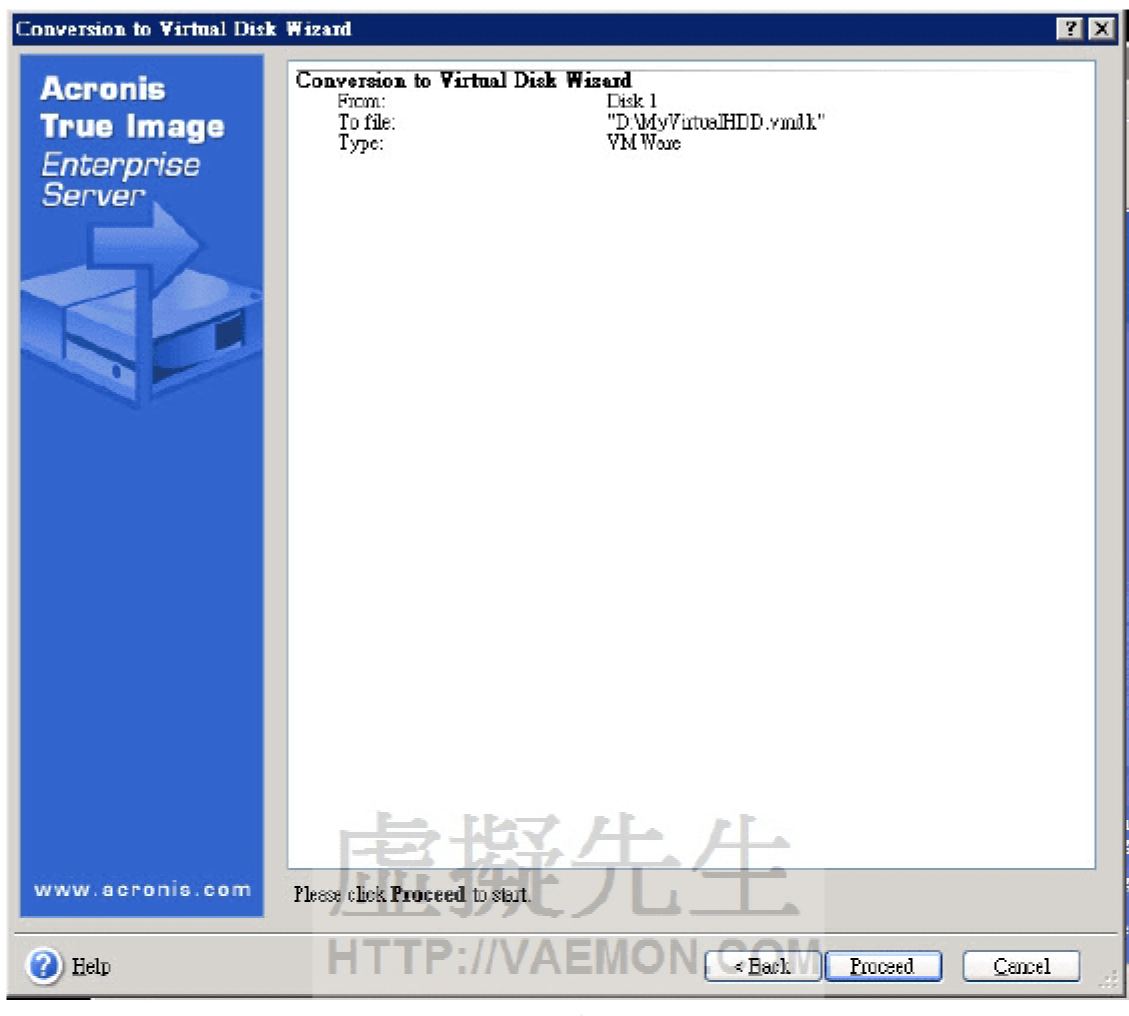
▲ 可以转换多种格式

5．接下来选择档名。单击 Next 按钮继续。



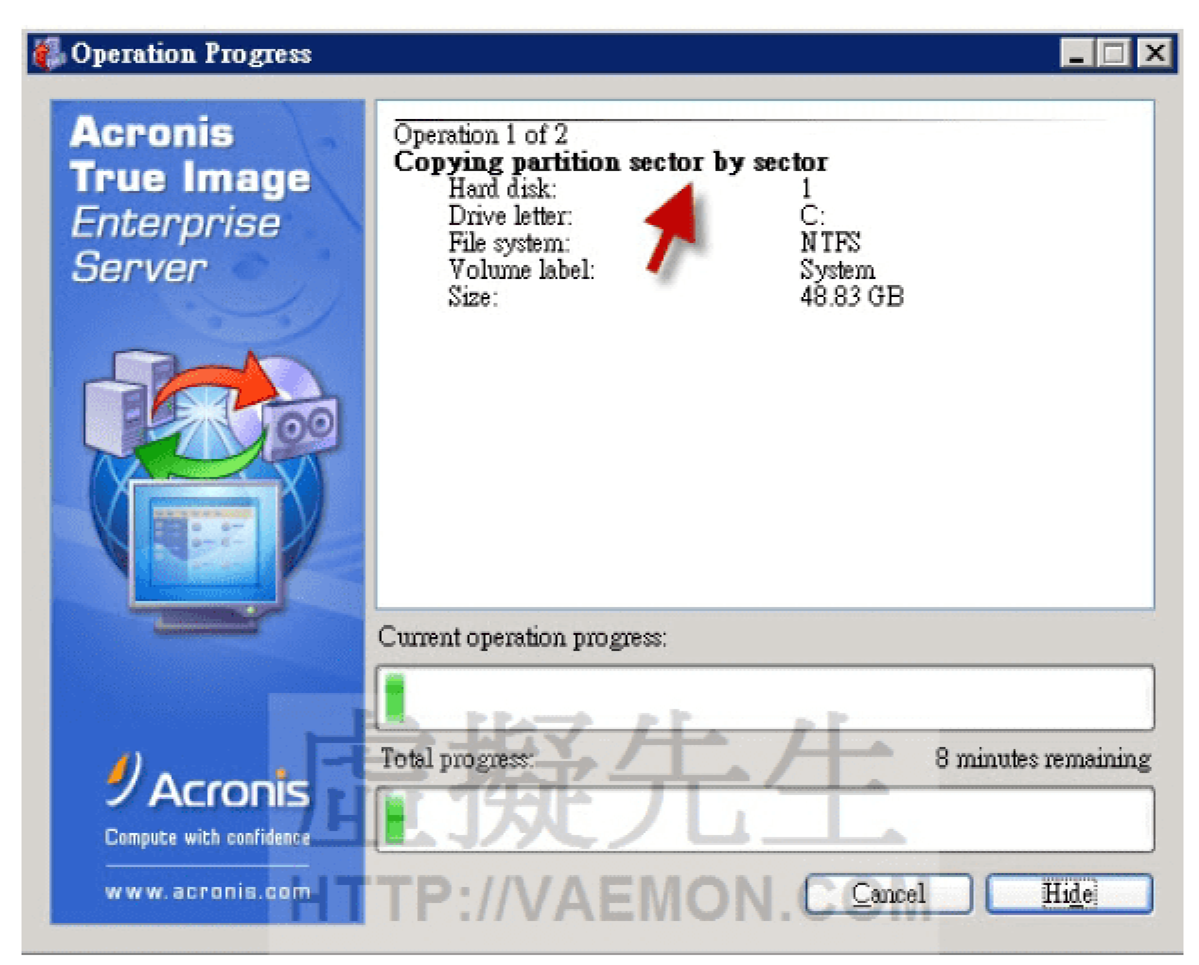
▲ 选择文件

6．单击 Proceed 按钮就开始了。



▲ 开始转换

7. 开始转换的画面，这里使用的是 sector by sector 的转换，所以会久一点。当这个画面结束，就意味着转换落实了。



▲ 转换时间会稍长一点

8. 由于转换之后，只有一个 VMDK，因此我们必须将这个 VMDK 上传到 ESX 的 Datastore 中，并且创建虚拟机。

9. 创建虚拟机这种简单的任务就不用写步骤，但别忘了创建时，要选择对应的操作系统，并且在创建新的虚拟磁盘时，要选择“使用现存的虚拟磁盘”，将刚才转换出来的 VMDK 加入虚拟机即可。

4. P2V 落实的任务

如果你选择的是 VMware，别忘了安装 VMware Tools，网卡、显示适配器之类的才能继续用。另外在转换成功之后，别忘了将原来的主机关闭，或是要将 IP 及主机名转换，要不然网卡是不可以有同 IP，而 Windows 网域中也是不可以有同名的主机的。

24.3.2 使用 WinImage 工具进行 P2V

P2V 的过程虽然耗时耗工，但如果你的物理机上的数据并不多，使用一些现存的小工具也是十分方便的。P2V 的热点就是要能完全转换，并且不能改变用户、系统管理员的基本习惯，最好是 IP，域名都不变。当然 P2V 还有一个热点是服务不中断。这一点也是一般公司面对 P2V 的最大挑战。

1. 在线及脱机转换

当前 P2V 有几个种类，一个是在线转换，即物理机（P）不用关机直接转成虚拟机（V），另一个就是文件转换，此时系统大部分要脱机（有些不需要，如 True Image），然后将硬盘的文件格式转成 VMDK 或是 VHD。

发现这两大阵营各有优缺点。在在线转换部分最有名的就是之前介绍的 True Image 还有 VMware 自身出的 vCenter Converter StandAlone 版本。但 VMware 的 Converter 在在线转换通常都无法一次成功。举例来说，用 VMware 的 vCenter Converter 在直接转换到 ESX 的主机上时，Windows 系统通常都无法引导（游标停着动也不动），必须先转换成 VMware Workstation 的格式再转换

ESX 的格式才能引导。而 True Image 的转换可以做在线，但必须先转成 True Image 的 Backup 格式 （tib），要经过两次转换，因此很多人也将脑筋动到脱机转换上。

2. 何谓脱机转换

脱机转换就是将物理机的 File System 转成 VMDK 或 VHD，通常你也需要一个硬盘的副本档（TIB 或是 GHO）。但笔者个人认为脱机转换最好用的还是 V2V。你可以将 Hyper-V 的 VHD 格式和 VMware 的 VMDK 格式互转，速度快又能整个转换，唯一的麻烦就是转换完后要自身创建一个虚拟机，但这不是什么大问题，如果用熟了甚至可以写一个 PowerShell 或是 VMware CLI 的 script 来自动生成。在这一小节，就来看看一个格式转换的好帮手 WinImage。

3. WinImage 介绍

WinImage 是一个付费软件，但可以先下载使用 30 天。分别支持 X86/X64/Itanium 三种平台。下载后不需要安装直接运行即可。我们在下面，就来看看如何用 WinImage 把 Hyper-V 的虚拟机硬盘 VHD 转换成 VMware 的 VMDK 格式。

> **注意** 使用 WinImage 的注意事项
>
> *在转换开始之前，千万不要忘了要将物理机网络的驱动程序拆除，要不然转换完之后的 Windows 会一直弹出 BSOD。*
>
> *为了兼容性，可以将 Hyper-V 的网卡、USB、FLOPPY 等设备都先卸除，当我们在 VMware 用的时候，可以先在虚拟机中创建这些设备，这样就会直接用使用 VMware 所附的虚拟设备。*

4. 开始将 Hyper-V 转换成 VMDK

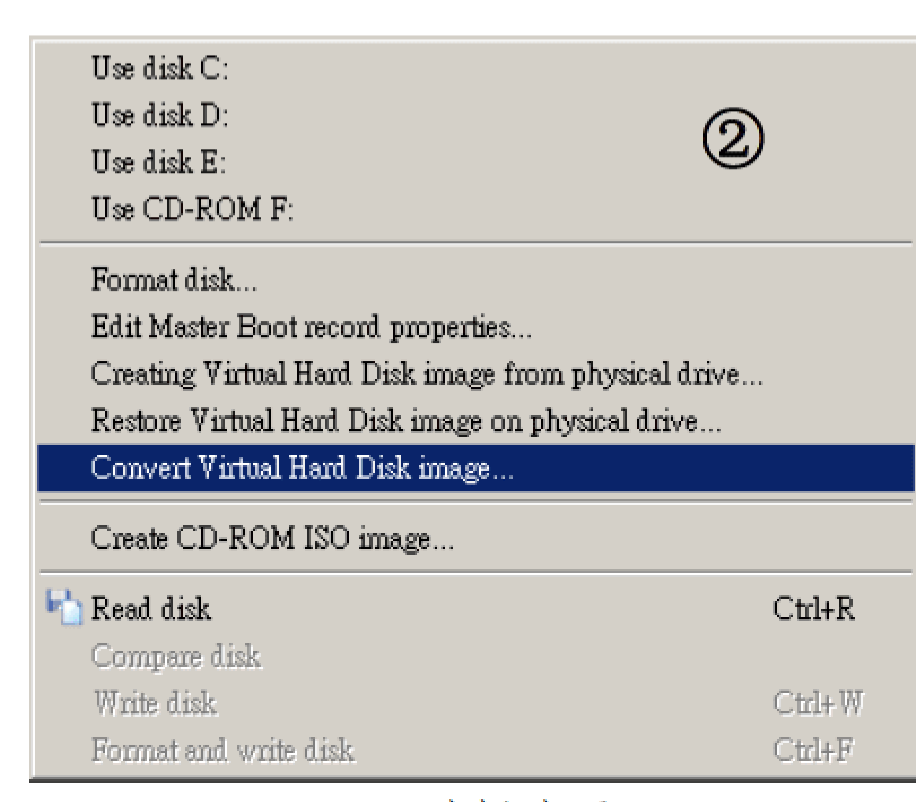
下面就是将微软 Hyper-V 格式的 VHD 转换成 VMware 格式 VMDK 的步骤。

▶ 将 VHD 转换成 VMDK

1. 首先下载并运行 WinImage。这里用的是 Windows 2008 R2，因此使用的是 64 比特版本的 WinImage。
2. 运行后选择 Disk/ Convert Virtual Hard Disk image 选项。



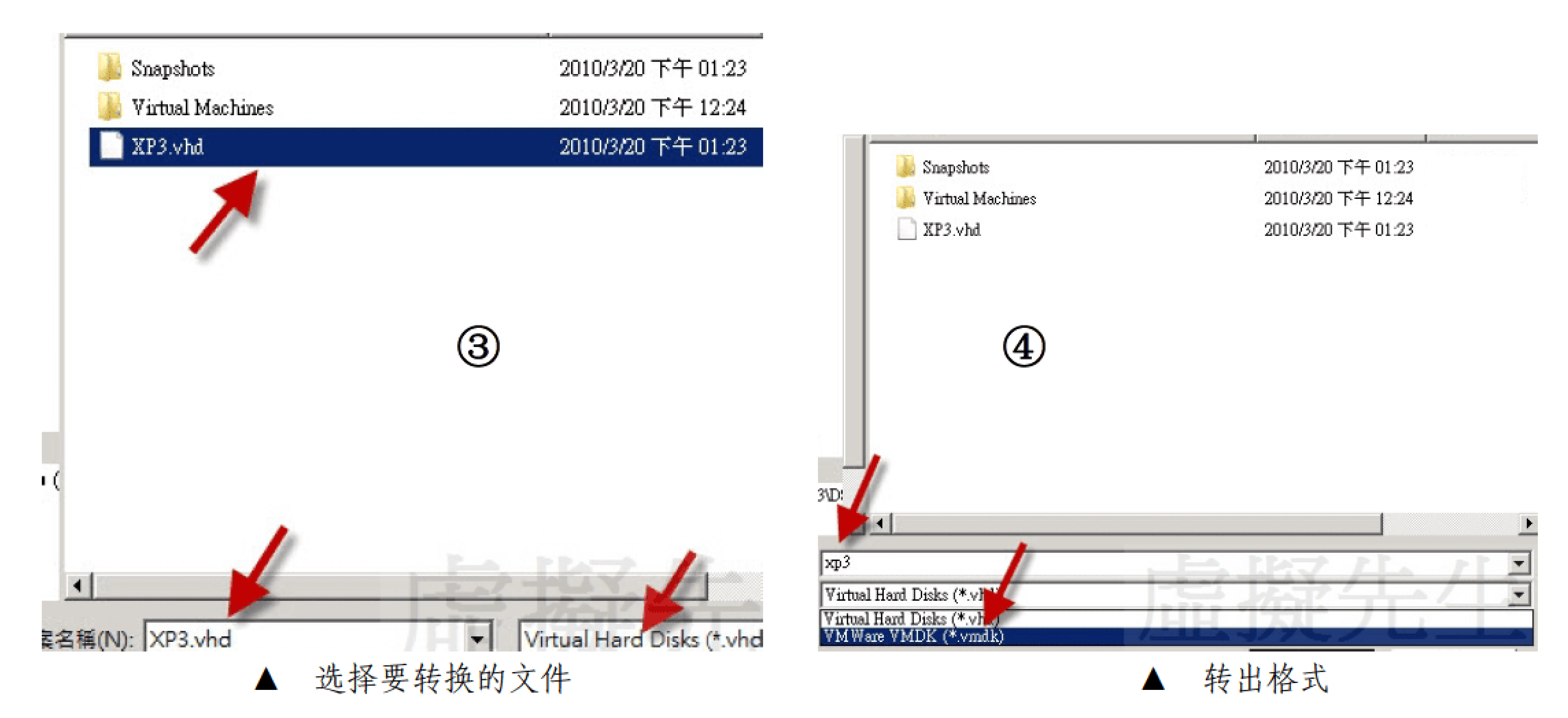
▲ 运行 WinImage



▲ 选择选项

3. 选择要转换的 Hyper-V 格式 VHD。WinImage 也可以反向转换。此时系统会询问是否要动态添加或是固化大小，一般选固化大小。

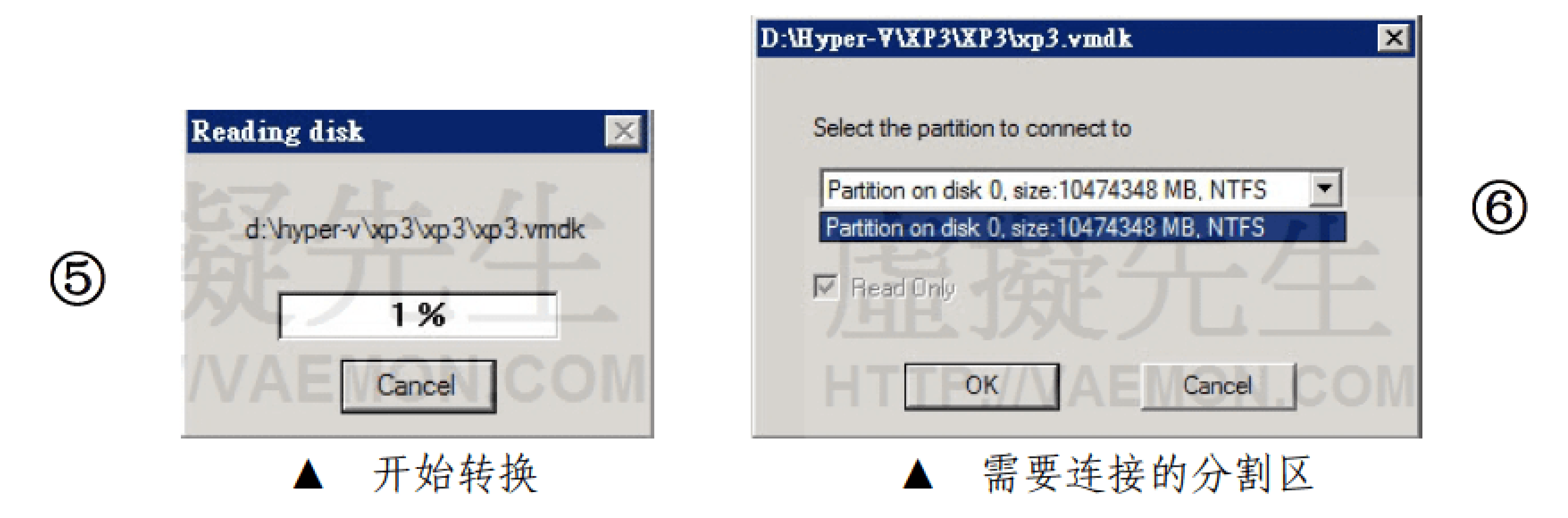
4. 接下来选择要转出的格式，我们这里选择 VMDK，并且选择输出的文件名。



▲ 选择要转换的文件　　▲ 转出格式

5. 接下来就会开始转换，大概数分钟到十小时不等，视你的 VHD 大小。

6. 转换完毕之后，系统会询问要连接到哪一个分区，我们这里只有一个分割区，就选择默认的。



▲ 开始转换　　▲ 需要连接的分割区

7. 接下来就进入 VMware 中创建一个新的 VM，细节就不详述了，但热点要选。

8. 在选择虚拟机的硬盘时，别忘了选择 Existing virtual disk。

9. WinImage 的好处，就是转换出来的 VMDK 也可以给 ESX/ESXi 运行。笔者试过 VMware 自身出的 vCenter Converter 转换出来的 VMware Workstation 格式，常常在配置有问题时，无法让 ESX 运行。

10. 如果是用 ESX/ESXi，别忘了选择这里，也是挂上已经创建好的 VMDK，别忘了要先上传到 Inventory。

结 语

所谓万事开头难，P2V 是企业虚拟化的第一步，但只要将物理机转上虚拟环境之后，就可以享受各式各样 vSphere 的功能，企业之后的 IT 管理也会越来越方便及顺利，因此 P2V 虽然会遇到许多问题，但绝对是最重要，也是最值得的一步。

第 25 章 用点命令处理 vSphere 的操作

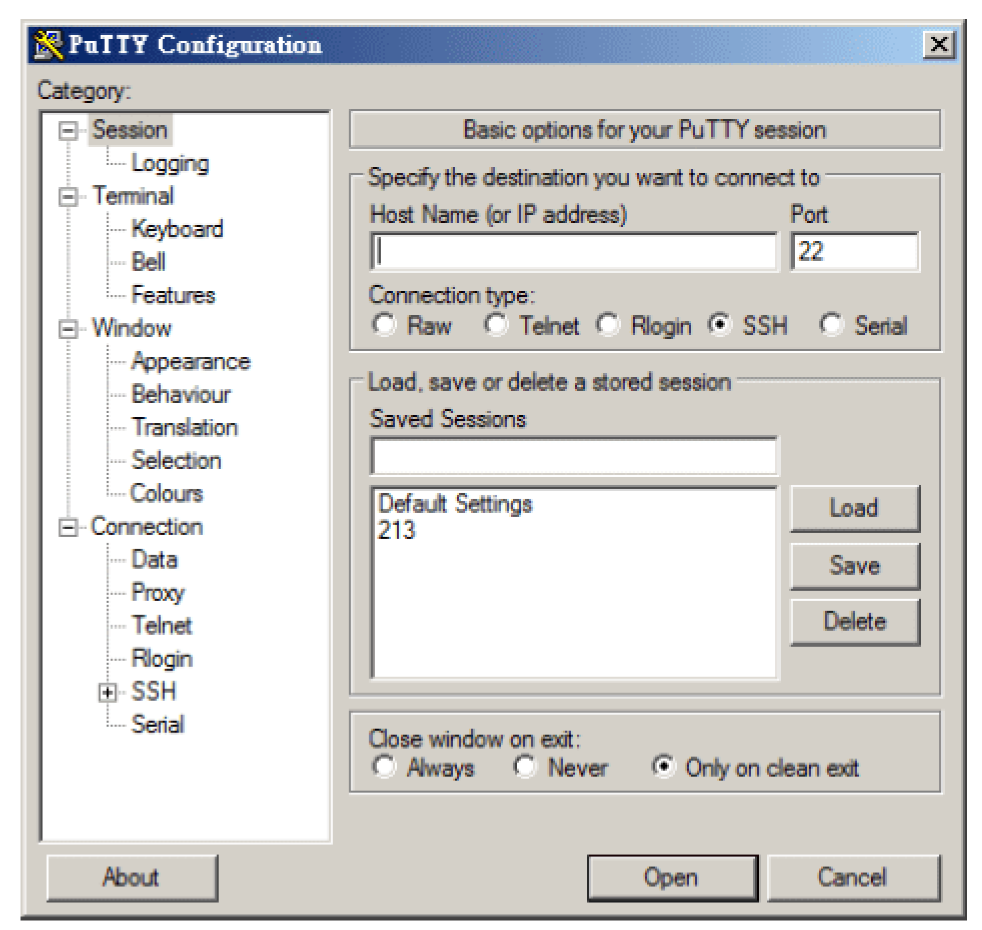
关键词：

- 理解 esxcfg-开头的命令
- 理解 vmware-开头的命令
- 理解 vmk 开头的命令
- 理解其他常用的命令
- 理解 ESX 中的配置文件

虽然 vSphere 的 vCenter 或 Client 已经能做到 90%操作虚拟机的任务，但总是有些时候，我们必须进入 ESX/ESXi 的 Console 中进行命令的操作。在本书的最后一章，我们就来看看最常用的 ESX/ESXi 命令。

25.1 最常用的命令

在开始之前，读者们别忘了先打开 ESX 主机上的 SSH，可以查阅第 9 章有关 SSH 打开的方式。本章所有的操作，都尽量集中在 SSH 的命令，因此大家也别忘了将 SSH 的 Client 端安装起来，也请引用第 9 章有关 PuTTY 的安装及使用。



▲ 使用 PuTTY 连入 ESX 主机

虽然大部分的时间都使用 vCenter 来管理虚拟机，但如果我们也能将 SSH 连入 ESX，在没有 vCenter 的情况也能处理一些紧急事件。下面就是常用的 ESX 命令清单，读者们可以在需要时随时查阅。

▶ 常用的 ESX ServCenterce Console 命令

- 给出当前网卡信息：esxcfg-vswif -l
- 检视 ESX 版本：vmware -v
- 给出 ESX 所打开的网络服务：esxcfg-firewall -s
- 检视某个服务是否打开：esxcfg-firewall -q SSHclinet（这里以 SSH 客户端为例）
- 重新装载 vmware 服务：servCenterce mgmt-vmware restart
- 给出当前所有虚拟交换机：esxcfg-vswitch -l
- 给出系统的网卡：esxcfg-nics -l

添加一个虚拟交换机，名称为 internal、连接到两片物理网卡。

- esxcfg-vswitch -a vSwitch1
- esxcfg-vswitch -A internal vSwitch1
- esxcfg-vswitch -L vmnic1 vSwitch1
- esxcfg-vswitch -L vmnic2 vSwitch1
- 删除交换机：esxcfg-vswitch -D vSwitch1
- 删除交换机上的网卡：esxcfg-vswitch -u vmnic1 vswitch2
- 删除 portgroup：esxcfg-vswitch -D internel vswitch1

创建 vmkernel switch。如果要用 VMotion 必须激活 iSCSI，此时需创建 portgroup。

- esxcfg-vswitch -l

- esxcfg-vswitch -a vswitch2
- esxcfg-vswitch -A "vm kernel" vswitch2
- esxcfg-vswitch -L vmnic3 vswitch2
- esxcfg-vmknic -a "vm kernel" -i 192.168.1.141 -n 255.255.255.0
- esxcfg-route 192.168.1.1

打开防火墙 SSH 端口：

- esxcfg-firewall -e SSHclient
- esxcfg-firewall -d SSHclient

创建总控台：

- esxcfg-vswitch -a vSwitch0
- esxcfg-vswitch -A "servCenterce console" vSwitch0
- esxcfg-vswitch -L vmnic0 vSwitch0
- esxcfg-vswif -a vswif0 -p "servCenterce console" -i 192.168.1.88 -n 255.255.255.0
- 添加 NAS 设备（其中 a 为添加选项卡，-o，是 NAS 服务器的名字或 IP，-s 是 NAS 键入的共享名称）：esxcfg-nas -a MYNAS -o 192.168.1.50 -s MYNAS
- 给出 NAS 连接：esxcfg-nas -l
- 强迫 ESX 连接 NAS 服务器：esxcfg-nas -r
- 检视连接 NAS 情况：esxcfg-nas -l
- 连接 iSCSI 设备（e:enable q:查询 d：disable s:强迫搜索）：esxcfg-swiscsi -e
- 设置 iSCSI 的目标 IP：vmkiscsi-tool -D -a 192.168.1.50 vmhba40（vmhba 是 iSCSI 的连接网络界面）
- 给出和 target 的连接：vmkiscsi-tool -l -T vmhba40
- 给出当前的 LUN：ls -l /vmfs/devCenterces/disks

25.1.1 使用 esxcfg 开头的命令

在 ESX 中，大部分命令都是以 esxcfg-开头，当你在 ESX 中键入 esxcfg-之后再按 Tab 键，就会弹出所有的命令清单，如果你再键入某一个命令的英文，系统就会自动将这个命令给出来。我们就以字母的排列的方式来说明命令。

1. esxcfg-addons

- 功能：给出 ESX 主机所安装的 Addons。

由于 ESX 会在引导时装载 Addons，才能达成 ESX 各种功能。这个命令是用来管理 ESX 的 Addons。使用手段也很简单，只要键入该命令，再加上参数，就可以使用。最常用的命令就是 list，可以给出当前的 Addons，当然也可以使用 Enable 或 Disable 来激活或停止某个 Addon。

```
[root@esxu2 ~]#
[root@esxu2 ~]# esxcfg-addons
usage: esxcfg-addons [action] [parameter(s)]
   help         display this help message
   list         list addons
   enable       enable disabled addon
   disable      disable enabled addon
   version      display version information
[root@esxu2 ~]#
```

▲ 运行 esxcfg-addon

2．esxcfg-advcfg

- 功能：进行 ESX 主机的配置。

```
root@esxu2:~
[root@esxu2 ~]# clear
[root@esxu2 ~]# esxcfg-advcfg
This usage
Usage: esxcfg-advcfg <options> [<adv cfg Path>]
   -g|--get                 Get the value of the VMkernel advanced
                            configuration option
   -s|--set <value>         Set the value of the VMkernel advanced
                            configuration option
   -d|--default             Reset a VMkernel advanced config option to default
   -q|--quiet               Suppress output
   -k|--set-kernel          Set a VMkernel load time option value for the next boot.
   -j|--get-kernel          Get a VMkernel load time option value for the next boot.
   -c|--get-config          Get the value of a specific line in esx.conf.
   -m|--set-message         Set DCUI welcome message.
   -u|--uuid                Ensure the Vmkernel system UUID is set and print it.
   -G|--get-user-var        Get the value of a user specified generic value.
                            Requires the --user-var option.
   -S|--set-user-var        Set the value of a user specified generic value.
                            Requires the --user-var option.
   -D|--del-user-var        Delete the value of user specified generic value.
                            Requires the --user-var option.
   -U|--user-var            The name of the user variable to user for
                            --get-user-var --set-user-var and --delete-user-var.
   -A|--add-option <name>   This adds an advanced variable to /UserVars tree of the
                            Advanced Options.  This is different from the
                            --user-var options, which do not show in the VCUI and
                            only contain simple user vars for use at boot time.
                            An advanced option requires the --add-type,
                            --add-desc, --add-default, --add-name and
                            --add-min, --add-max if type is int
   -T|--add-type            The type of user advanced option to add,
                            must be either 'string' or 'int'
   -E|--add-desc            The description of the user advanced option to add
```

▲ 运行 esxcfg-advcfg

esxcfg-advcfg 是标准的主机功能配置，也是 ESX 中说明码件最多的命令。主要的功能就是进行 ESX 主机的进阶配置。一般来说我们最常用的两个参数就是-g 和-s，其中-g 就是 get，获取系统信息；而“-s”就是 set，用来配置系统参数。举例来说，我们要获取主机名，就键入：

▶ 获取 ESX 主机名

```
[root@vaemon.com]# esxcfg-advcfg -g /Misc/HostName
```

```
root@esxu2:~
[root@esxu2 ~]#
[root@esxu2 ~]# esxcfg-advcfg -g /Misc/HostName
Value of HostName is esxu2
[root@esxu2 ~]#
```

▲ 运行结果

esxcfg-advcfg 的参数都列在/proc/vmware/config 之下，你可以从中来配置 ESX 的参数，大部分的参数都在子目录下，因此理解子目录就可以配置了。

▶ esxcfg-advcfg 的目录

- BufferCache
- CPU
- Disk
- FileSystem
- Irq
- LVM
- Mem
- Migrate
- Misc

- Net
- NFS
- Numa
- Scsi
- User
- VMFS3

如果你想给出所有的子目录，可以键入 esxcfg-info –o 这个命令。在 ESX 中，esxcfg-advcfg 最常用的就是配置存储设备以及虚拟硬盘。也可以配置硬盘的格式，如“默认填入 0”或是 Thin provisioning、LUN、iSCSI 等都是在此配置。

下面是几个范例程式码：

▶ esxcfg-advcfg 范例代码

```
[root@vaemon.com]# esxcfg-advcfg -s 128 /Disk/MaxLUN
```

```
[root@esx01 Disk]# esxcfg-advcfg -s 128 /Disk/MaxLUN
Value of MaxLUN is 128
[root@esx01 Disk]# _
```

▲ 运行结果

- 是否支持非 Thin Provisioning 的 ZeroedThick 虚拟磁盘。
- [root@vaemon.com]# esxcfg-advcfg –g /Disk/MaxLUN
- 支持最大的 LUN 数目。

```
[root@esx01 Disk]# esxcfg-advcfg -g /Disk/MaxLUN
Value of MaxLUN is 256
[root@esx01 Disk]# _
```

▲ 运行结果

3. esxcfg-auth

- 功能：配置 ESX 的授权及口令策略。

```
root@esxu2:/vmfs/volumes
[root@esxu2 volumes]# esxcfg-auth
usage: esxcfg-auth [options]

options:
  -h, --help            show this help message and exit
  --enablemd5           Enable MD5 password storage
  --disablemd5          Disable MD5 password storage
  --enableshadow        Enable Shadow password storage
  --disableshadow       Disable Shadow password storage
  --enablenis           Enable NIS Authentication
  --disablenis          Disable NIS Authentication
  --cleanupnis          Cleanup yp.conf and disable NIS Authentication
  --nisdomain=domain    Set the NIS domain
  --nisserver=server    Set the NIS server
  --nisypserver=server  Set the NIS ypserver
  --enableldap          Enable LDAP User Management
  --disableldap         Disable LDAP User Management
  --enableldapauth      Enable LDAP Authentication
  --disableldapauth     Disable LDAP Authentication
  --ldapserver=server   Set the LDAP Server
  --ldapbasedn=basedn   Set the base DN for the LDAP server
  --enableldaptls       Enable TLS connections for LDAP
  --disableldaptls      Disable TLS connections for LDAP
  --enablekrb5          Enable Kererbos Authentication
  --disablekrb5         Disable Kererbos Authentication
  --krb5realm=domain    Set the Kerberos Realm
  --krb5kdc=server      Set the Kebreros Key Distribution Center
  --krb5adminserver=server
                        Set the Kerberos Admin Server
  --enablead            Enable Active Directory Authentication
  --disablead           Disable Active Directory Authentication
  --addomain=domain     Set the Active Directory Domain
  --addc=server         Set the Active Directory Domain Controller
```

▲ 配置口令策略

配置 ESX service console 的用户权力，包括了 NIS、LDAP、Kerberos 和 Windows 的 AD。可以使用这个命令来让 vCenter 加入 AD，也可以配置口令的策略。举例来说，如果要配置 Windows

AD 在 vaemon.com 的网域，就要键入：

▶ 配置 AD 网域的 ESX 权限

```
[root@vaemon.com]# esxcfg-auth --enablead --addomain=vaemon.com --addc=
dc1.vaemon.com
```

```
[root@esxu2 volumes]# esxcfg-auth --enablead --addomain=vaemon.com --addc=dc1.vaemon.c
om
[root@esxu2 volumes]#
```

▲ 运行结果

如果要配置口令策略，让 90 天内就过期，60 天之后会有报警消息，至少要使用 30 天的话，就键入：

▶ 口令策略配置

```
[root@vaemon.com]# esxcfg-auth --passmaxdays=90 --passminidays=30
--passwarnage=60
```

```
root@esxu2:/vmfs/volumes
[root@esxu2 volumes]#
[root@esxu2 volumes]#  esxcfg-auth --passmaxdays=90 --passmindays=30 --passwarnage=60
[root@esxu2 volumes]#
```

▲ 运行结果

4．esxcfg-boot

■ 功能：用来给定 Linux GRUB 引导管理 ESX 主机。

```
root@esxu2:/vmfs/volumes
[root@esxu2 volumes]#
[root@esxu2 volumes]# esxcfg-boot
esxcfg-boot -h --help
            -f --force
            -b --update-bootcfg
            -t --update-trouble

   Queries cannot be combined with each other or other options.
[root@esxu2 volumes]#
```

▲ 给定引导

大家都知道 Linux 使用了 GRUB 来管理引导，这个命令就是用来管理 ESX 的引导的菜单。在 ESX4 之后 esxcfg-boot 的功能减少很多，主要还是通过 vCenter 来进行管理。

5．esxcfg-configcheck

■ 功能：遍历 ESX 的配置。

当前这个命令已停用，在 ESX 中暂时无法使用。

6．esxcfg-dumppart

■ 功能：用来配置 VMkernel 蓝屏时的倾印分割区。

当前这个命令只用来检视 ESX 主机的倾印分区是否已创建，键入下面的代码：

▶ 给出倾印的分割区

```
root@esxu2:/proc/vmware
[root@esxu2 vmware]# clear
[root@esxu2 vmware]#
[root@esxu2 vmware]# esxcfg-dumppart
esxcfg-dumppart <options> [<partition>]
-l|--list                List the partitions available for Dump Partitions.
                         WARNING: This will scan all LUNs on the system.
-t|--get-active          Get the active Dump Partition for this system,
                         returns the device identifier for the partition
                         (naa.xxxxx:1) or 'none'.
-c|--get-config          Get the configured Dump Partition for this
                         system, returns the device identifier for the partition
                         (naa.xxxxx:1) or 'none'.
-s|--set                 Set the Dump Partition for this system and activate it,
                         either naa.xxxxx:1 or 'none' to deactivate
                         the active dump partition.
-f|--find                Find usable Dump partitions and list in order of
                         preference.
-C|--copy                Copy a dump file from the dump partition:
                         -D --devname /vmfs/devices/disks/<partition name>
                         -z --zdumpname outputName (optional)
                         -n --newonly (optional)
                         -o --slot slotnum (optional)
                         -U --unqualified-dev (FOR INTERNAL USE ONLY)
-L|--log                 Extract  the log from a compressed dump.
-S|--smart-activate      Activate the configured dump partition or find
                         the first appropriate partition and use it
                         (same order as -f).
-a|--activate            Activate the configured dump partition.
-d|--deactivate          Deactivate the active dump partition.
-h|--help                Show this message.

Example:
```

▲ 倾印分区

- esxcfg-dumppart –l

```
    esxcfg-dumppart --log esxdump
[root@esxu2 vmware]# esxcfg-dumppart -l
VM Kernel Name                                          Console Name
    Is Active   Is Configured
mpx.vmhba0:C0:T0:L0:2                                   /dev/sda2
    yes            yes
[root@esxu2 vmware]#
```

▲ 运行结果

7. esxcfg-firewall

- 功能：配置 ESX 主机上的网络防火墙

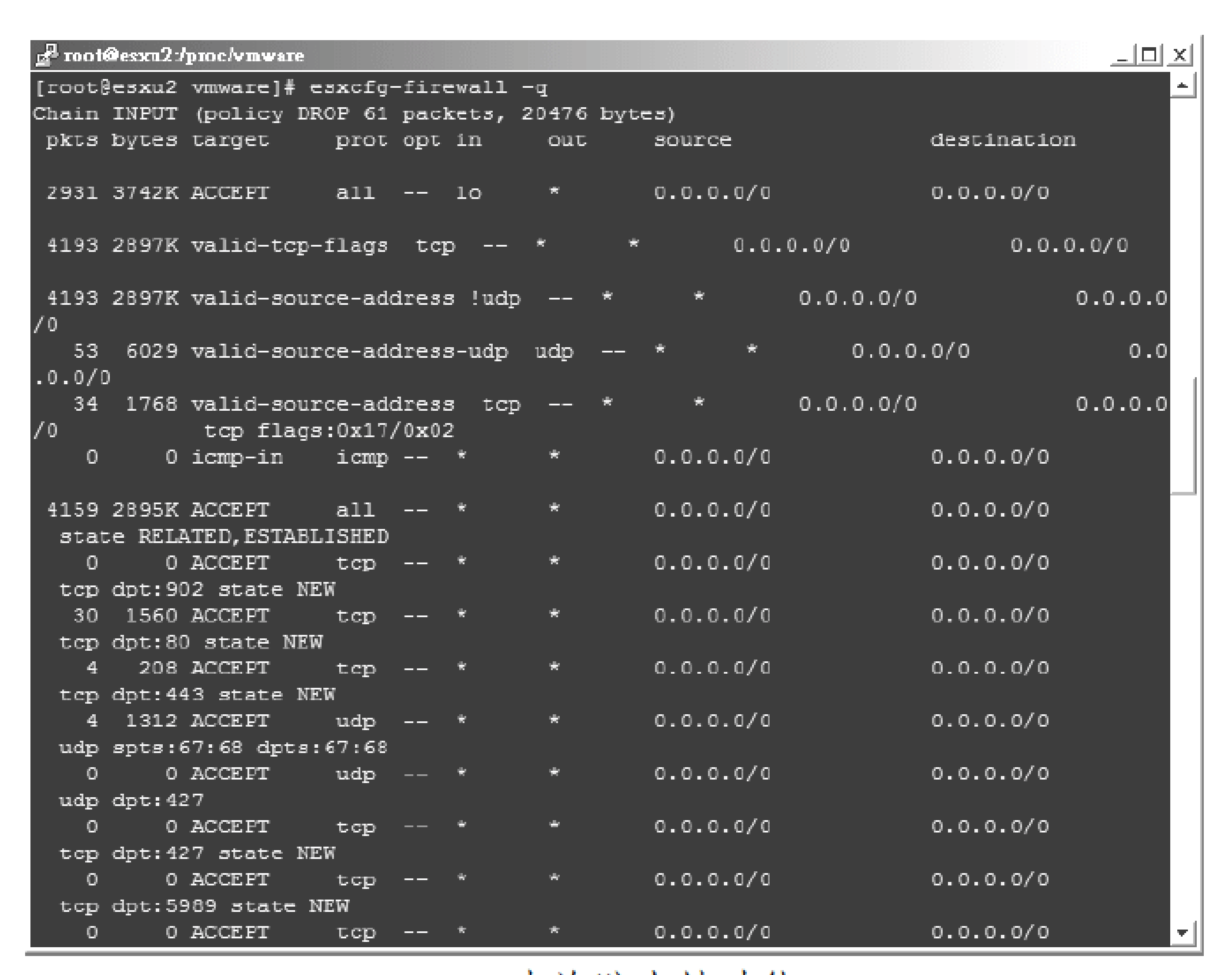
```
root@esxu2:/proc/vmware
[root@esxu2 vmware]#
[root@esxu2 vmware]# esxcfg-firewall
esxcfg-firewall <options>
-q|--query                                    Lists current settings.
-q|--query <service>                          Lists setting for the
                                              specified service.
-q|--query incoming|outgoing                  Lists setting for non-required
                                              incoming/outgoing ports.
-s|--services                                 Lists known services.
-l|--load                                     Loads current settings.
-r|--resetDefaults                            Resets all options to defaults
-e|--enableService <service>                  Allows specified service
                                              through the firewall.
-d|--disableService <service>                 Blocks specified service
-o|--openPort <port,tcp|udp,in|out,name>      Opens a port.
-c|--closePort <port,tcp|udp,in|out>          Closes a port previously opened
                                              via --openPort.
   --ipruleAdd <host,cport,tcp|udp,REJECT|DROP|ACCEPT,name>  Adds a rule
                                              to block/allow hosts to access
                                              specific COS service;'cport' can
                                              be specified like 'a:b'. For ex:
                                              0:65535 blocks all the ports;
                                              'host' can specified like 'a/b'.
                                              For ex: 0.0.0.0/0 blocks all the
                                              hosts.
   --ipruleDel <host,cport,tcp|udp,REJECT|DROP|ACCEPT>    Deletes the host rule
                                              previously added via --ipruleAdd
   --moduleAdd <module>                       Loads an iptables module, and
                                              adds it to the peristent
                                              firewall configuration.
   --moduleDel <module>                       Removes an iptables module, and
                                              removes it from the persistent
                                              firewall configuration.
```

▲ 配置防火墙

配置 ESX 上的防火墙是否让服务进入或出去。由于 ESX 的 Service Console 是由 RedHat Linux

改的，而 Linux 的防火墙是由 iptables 来总控。但 iptables 很难用，因此 VMware 特别开发了这个命令方便大家来操作。Esxcfg-firewall 的配置是存储在/etc/vmware/esx.conf 中，但我们不会直接编这个文件。如果在配置时弹出问题，则是检视/var/log/vmware/esxcfg-firewall.log。

esxcfg-firewall 这个命令最常用的参数就是－q，就是查询的意思，但你必须给定要查询的参数。另一个常用的就是－s，用来激活或是关闭某些功能。下面就是最常用的功能。这些功能都定义在/etc/vmware/firewall/services.xml 中。



```
root@esxu2:/proc/vmware
[root@esxu2 vmware]# esxcfg-firewall -q
Chain INPUT (policy DROP 61 packets, 20476 bytes)
 pkts bytes target     prot opt in     out     source               destination

 2931 3742K ACCEPT     all  --  lo     *       0.0.0.0/0            0.0.0.0/0

 4193 2897K valid-tcp-flags  tcp  --  *      *       0.0.0.0/0            0.0.0.0/0

 4193 2897K valid-source-address !udp  --  *      *       0.0.0.0/0            0.0.0.0
/0
   53  6029 valid-source-address-udp  udp  --  *      *       0.0.0.0/0            0.0
.0.0/0
   34  1768 valid-source-address  tcp  --  *      *       0.0.0.0/0            0.0.0.0
/0           tcp flags:0x17/0x02
    0     0 icmp-in    icmp --  *      *       0.0.0.0/0            0.0.0.0/0

 4159 2895K ACCEPT     all  --  *      *       0.0.0.0/0            0.0.0.0/0
  state RELATED,ESTABLISHED
    0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0
  tcp dpt:902 state NEW
   30  1560 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0
  tcp dpt:80 state NEW
    4   208 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0
  tcp dpt:443 state NEW
    4  1312 ACCEPT     udp  --  *      *       0.0.0.0/0            0.0.0.0/0
  udp spts:67:68 dpts:67:68
    0     0 ACCEPT     udp  --  *      *       0.0.0.0/0            0.0.0.0/0
  udp dpt:427
    0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0
  tcp dpt:427 state NEW
    0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0
  tcp dpt:5989 state NEW
    0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0
```

▲ 查询防火墙功能

▶ esxcfg-firewall 最常用的功能查询

- nfsClient
- ftpServer
- ntpClient
- dellom
- nisClient
- vncServer
- tmpLicenseClient
- swISCSIClient
- CIMHttpsServer
- sshClient
- snmpd
- tmpAAMClient
- vpxHeartbeats
- smbClient
- hpim
- tmpHostVmdbServer
- tmpHostdSOAPServer
- ftpClient

- sshServer
- ibmdirector
- CIMHttpServer
- telnetClient

在这个命令中，一般使用—e 来激活某个已知的服务，使用—d 来关闭已知的服务。举例来说，如果我们要激活从 ESX 主机至其他主机的 SSH 客户端服务，就需要键入下面的命令：

▶ 激活 SSH 客户端

```
[root@vaemon.com]# esxcfg-firewall -e sshClient
```

```
root@esxu2:/proc/vmware
[root@esxu2 vmware]#
[root@esxu2 vmware]# esxcfg-firewall -e sshClient
[root@esxu2 vmware]#
```

▲ 运行结果

如果要将某个服务关闭，则是使用—d 参数。以上面为例，如果要关闭 SSH 的话，则键入：

▶ 关闭 SSH 客户端

```
[root@vaemon.com]# esxcfg-firewall -d sshClient
```

```
root@esxu2:/proc/vmware
[root@esxu2 vmware]#
[root@esxu2 vmware]# esxcfg-firewall -e sshClient
[root@esxu2 vmware]# esxcfg-firewall -d sshClient
[root@esxu2 vmware]#
```

▲ 运行结果

当然也可以给定 TCP 或 UDP 或是通信端口号码，这是使用—o 参数进行。举例来说，如果我们要使用 MySQLclient 功能，并且使用 TCP 3306 的 out 端，则可以键入：

▶ 开放 MySQL 客户端

```
[root@vaemon.com]# esxcfg-firewall -o 3306,tcp,out,MySQLclient
```

```
root@esxu2:/proc/vmware
[root@esxu2 vmware]#
[root@esxu2 vmware]# esxcfg-firewall -o 3306,tcp,out,MySQLclient
[root@esxu2 vmware]#
```

▲ 运行结果

8. esxcfg-hwiscsi

- 功能：用来配置硬件 iSCSI 的参数。

esxcfg-hwscsi 是用来配置硬件 iSCSI 卡的参数，主要是配置是否要用 Jumbo Frame 等参数，或是卡上的 ARP 转向。

```
[root@esxu2 vmware]#
[root@esxu2 vmware]# esxcfg-hwiscsi
Incorrect number of arguments.
Usage: /usr/sbin/esxcfg-hwiscsi [-l] [-a allow|deny] [-j enable|disable] [-h] <vmkerne
l SCSI adapter name>
  -l: list current configuration (overrides setting options)
  -a: allow or deny ARP redirection on adapter
  -j: enable or disable jumbo frame support
      (enabled: MTU = 9000, disabled: MTU = 1500)
  -h: print this message
[root@esxu2 vmware]#
```

▲ esxcfg-hwiscsi

9. esxcfg-info

- 功能：显示 ESX 主机的各项信息。

```
            |----Default Value.......................................false
         \==+Kernel Bool Option :
            |----Option Name.........................................rollback
            |----Configured Value....................................false
            |----Runtime Value.......................................false
            |----Default Value.......................................false
         \==+Kernel Uint8 Option :
            |----Option Name.........................................serialPort
            |----Configured Value....................................0
            |----Runtime Value.......................................0
            |----Default Value.......................................0
         \==+Kernel Bool Option :
            |----Option Name.........................................sharePerNode
            |----Configured Value....................................true
            |----Runtime Value.......................................true
            |----Default Value.......................................true
         \==+Kernel Bool Option :
            |----Option Name.........................................skipMicrocodeCompa
tCheck
            |----Configured Value....................................false
            |----Runtime Value.......................................false
            |----Default Value.......................................false
         \==+Kernel Bool Option :
            |----Option Name.........................................smallFontForTTY
            |----Configured Value....................................false
            |----Runtime Value.......................................false
            |----Default Value.......................................false
         \==+Kernel Uint32 Option :
            |----Option Name.........................................storageHeapMaxSize
            |----Configured Value....................................0
            |----Runtime Value.......................................0
            |----Default Value.......................................0
```

▲ esxcfg-info

这个命令的用处就是显示 ESX 主机的各项信息，主要让你理解当前有哪些服务在进行中，哪些服务停止了。如果要直接运行没有参数的命令，最好将结果导入一个文本档，要不然会有太多信息。一般来说，这个命令分成六个开关，使用不同的英文字母代表，因此加入开关可以检视不同的大类。这些开关分别如下。

▶ esxcfg-info 的六大类

- w　hardware
- r　resource
- s　storage
- n　network
- y　system
- advanced options

10. esxcfg-module

- 功能：配置 ESX 激活时装载驱动程序的参数。



▲ esxcfg-module

ESX 在激活时会装载驱动程序，在 ESX 中称为 Modules。这且可以检视或是配置装载 VMKernel 模块的数。最常用的就是以–l 来检视参数。一般来说，当我们想要修改 VMkernel 模块的功能时，就会使用这个命令。使用的命令如下：

▶ **修改 HBA 卡的参数**

```
[root@vaemon.com]# esxcfg-module -l
```

```
[root@esxu2 /]# esxcfg-module -l
Name                 R/O Addr        Length      R/W Addr        Length     ID
    Loaded
vmkernel             0x41803ca00000  0x29fda5    0x0             0x0        0
    Yes
vmklinux             0x41803cd80000  0x64000     0x417ffdcb8500  0x31000    1
    Yes
random               0x41803cde4000  0x3000      0x417ffdcf1dc0  0x4000     2
    Yes
dm                   0x41803cde7000  0x2000      0x417ffdcf5ed0  0x1000     3
    Yes
nmp                  0x41803cde9000  0x18000     0x417ffdcf6ee0  0x2000     4
    Yes
vmw_satp_local       0x41803ce01000  0x1000      0x417ffdcf8f30  0x1000     5
    Yes
vmw_satp_default_aa  0x41803ce02000  0x1000      0x417ffdcf9f40  0x1000     6
    Yes
vmw_satp_alua        0x41803ce03000  0x4000      0x417ffdcfaf80  0x1000     7
    Yes
vmw_psp_fixed        0x41803ce07000  0x2000      0x417ffdcfbfa0  0x1000     8
    Yes
libata               0x41803ce09000  0x15000     0x417ffdcfcfe0  0x2000     9
    Yes
mask_path_plugin     0x41803ce1e000  0x1000      0x417ffdcff810  0x1000     a
    Yes
vmw_satp_cx          0x41803ce1f000  0x6000      0x417ffdd00820  0x1000     b
    Yes
vmw_satp_default_ap  0x41803ce25000  0x2000      0x417ffdd01830  0x1000     c
    Yes
vmw_satp_eva         0x41803ce27000  0x2000      0x417ffdd02840  0x1000     d
    Yes
```

▲ 运行结果

11. esxcfg-mpath

- 功能：配置 ESX 主机上存储设备的多重路径。

```
root@esxu2:/
[root@esxu2 /]#
[root@esxu2 /]# esxcfg-mpath
esxcfg-mpath <options> [--path=<path>]
-l|--list                     List all Paths on the system with their
                              detailed information.
-L|--list-compact             List all Paths with abbreviated information
-m|--list-map                 List all Paths with adapter and device mappings.
-b|--list-paths               List all devices with their corresponding paths.
-G|--list-plugins             List all Multipathing Plugins loaded into the
                              system.
-s|--state <active|off>       Set the state for a specific LUN Path.  Requires
                              path UID or path Runtime Name in --path
-P|--path                     Used to specify a specific path for operations.
                              The path name may be either the long Path UID or
                              the shorter runtime name of the path.  This can
                              be used to filter any of the list commands to
                              a specific path if applicable.
-d|--device                   Used to filter the list commands to display
                              only a specific device.
-r|--restore                  Restore path setting to configured values
                              on system start. (INTERNAL USE ONLY)
-h|--help                     Show this message.


[root@esxu2 /]#
```

▲ esxcfg-mpath

ESX 主机在接上外置存储设备时，为了防止单点失误，需要配置多个路径，称为多重路径（Multipath）。本命令就是这个功能。首先我们会利用—l 来给出存储和路径，命令如下：

▶ 给出多重路径

```
[root@esx1host tools-isoimages]# esxcfg-mpath -l
```

```
[root@esxu2 /]# esxcfg-mpath -l
ide.vmhba32-ide.0:0-mpx.vmhba32:C0:T0:L0
   Runtime Name: vmhba32:C0:T0:L0
   Device: mpx.vmhba32:C0:T0:L0
   Device Display Name: Local NECVMWar CD-ROM (mpx.vmhba32:C0:T0:L0)
   Adapter: vmhba32 Channel: 0 Target: 0 LUN: 0
   Adapter Identifier: ide.vmhba32
   Target Identifier: ide.0:0
   Plugin: NMP
   State: active
   Transport: ide

pscsi.vmhba0-pscsi.0:0-mpx.vmhba0:C0:T0:L0
   Runtime Name: vmhba0:C0:T0:L0
   Device: mpx.vmhba0:C0:T0:L0
   Device Display Name: Local VMware, Disk (mpx.vmhba0:C0:T0:L0)
   Adapter: vmhba0 Channel: 0 Target: 0 LUN: 0
   Adapter Identifier: pscsi.vmhba0
   Target Identifier: pscsi.0:0
   Plugin: NMP
   State: active
   Transport: parallel

[root@esxu2 /]#
```

▲ 运行结果

12．esxcfg-nas

- 功能：配置 ESX 中的外部存储，主要是 NFS 的处理。

```
[root@esxu2 /]# esxcfg-nas
esxcfg-nas <options> [<label>]
-a|--add                Add a new NAS filesystem to /vmfs volumes.
                        Requires --host and --share options.
                        Use --readonly option only for readonly access.
-o|--host <host>        Set the host name or ip address for a NAS mount.
-s|--share <share>      Set the name of the NAS share on the remote system.
-y|--readonly           Add the new NAS filesystem with readonly access.
-d|--delete             Unmount and delete a filesystem.
-l|--list               List the currently mounted NAS file systems.
-r|--restore            Restore all NAS mounts from the configuration file.
                        (FOR INTERNAL USE ONLY).
-h|--help               Show this message.
[root@esxu2 /]#
```

▲ esxcfg-nas

esxcfg-nas 主要是用来给出 NFS，并且也可以挂载或是卸除 NFS 硬盘，主要是处理外部存储的 NFS。最常用的命令就是—l 给出所有的 NFS 硬盘，你可以键入以下的命令来给出所有的外部网络存储设备，但不会给出 iSCSI 的设备。

▶ 给出 esxcfg-nas 的命令

- Esxcfg-nas –l

如果你想要挂载一个 NFS，只要键入下面的命令即可。

▶ 挂载 NFS

```
esxcfg-nas -a -o 192.168.1.100 -s /NFS NFS01
```

其中 192.168.1.100 是 NFS 的 IP，NFS01 是该 NFS 的名称。一般来说，格式为 NFS，一般会将这个 NFS 挂上，并且创建连接。如果要创建一个 NFS 的连接，当然需要 IP 地址，作为这个 NFS 的 Client 端，此时必须要给这个 VMKernel 一个通信端口创建在虚拟交换机上，此时必须使用 esxcfg-vmknic，我们会在下面的章节说明。

13．esxcfg-pciid

- 功能：给出 ESX 主机上的 PCI 设备的 XML 文件。

```
root@esxu2:/
kernel driver mapping for device id 105a:4d69 in /etc/vmware/pciid/pata_pdc2027x.xml c
onflicts with definition for unknown
kernel driver mapping for device id 105a:5275 in /etc/vmware/pciid/pata_pdc2027x.xml c
onflicts with definition for unknown
kernel driver mapping for device id 105a:6268 in /etc/vmware/pciid/pata_pdc2027x.xml c
onflicts with definition for unknown
kernel driver mapping for device id 105a:6269 in /etc/vmware/pciid/pata_pdc2027x.xml c
onflicts with definition for unknown
kernel driver mapping for device id 105a:7275 in /etc/vmware/pciid/pata_pdc2027x.xml c
onflicts with definition for unknown
loading /etc/vmware/pciid/pata_serverworks.xml...
loading /etc/vmware/pciid/pata_sil680.xml...
loading /etc/vmware/pciid/qla2xxx.xml...
loading /etc/vmware/pciid/qla4xxx.xml...
loading /etc/vmware/pciid/random.xml...
loading /etc/vmware/pciid/s2io.xml...
loading /etc/vmware/pciid/sata_nv.xml...
loading /etc/vmware/pciid/sata_promise.xml...
loading /etc/vmware/pciid/sata_sil.xml...
loading /etc/vmware/pciid/sata_svw.xml...
loading /etc/vmware/pciid/tg3.xml...
loading /etc/vmware/pciid/tpm_tis.xml...
loading /etc/vmware/pciid/usb-ohci.xml...
loading /etc/vmware/pciid/usb-storage.xml...
loading /etc/vmware/pciid/usb-uhci.xml...
loading /etc/vmware/pciid/usb.xml...
writing /etc/vmware/pci.xml.merged
Replacing /etc/vmware/pci.ids file...
Replacing /etc/vmware/pcitable file...
Replacing /etc/vmware/pcitable.Linux file...
Replacing /etc/vmware/vmware-devices.map file...
Replacing /etc/vmware/simple.map file...
[root@esxu2 /]#
```

▲ esxcfg-pciid

ESX 的主机使用的 PCI 设备会使用 XML 文件来定义，这个命令会给出所有的 PCI 设备，当

设备弹出补丁时，会自动来对应 XML 文件并且补丁。

14．esxcfg-rescan

- 功能：用来重新扫描 HBA 卡上的存储设备。

当我们在外部存储设备上定义了新的 LUN 时，必须重新扫描，在 vSphere Client 中使用 rescan 的功能，就是这个命令。如果我们使用了 iSCSI 的存储卡，其名称为 vmhba32，那么要重新扫描这个卡上的设备，只要键入：

▶ **重新扫描 HBA**

```
[root@vaemon.com]# esxcfg-rescan vmhba32
```

15．esxcfg-resgrp

- 功能：管理 ESX 中的资源群组（Resource Group）。

```
root@esxu2:/
[root@esxu2 /]#
[root@esxu2 /]# esxcfg-resgrp
Usage: esxcfg-resgrp [options] [resgrp path]
  -a, --add rgname     Add a new group under the path
  -d, --delete         Delete the resource group
  -l, --list           List all the resource groups
                       Or stats about a specific resource group
      --restore        Restore all resource groups from configuration
                       file (FOR INTERNAL USE ONLY).
  -h, --help           Show this message
[root@esxu2 /]#
```

▲ esxcfg-resgrp

在 ESX 中，可能将许多资源分配成资源组，这个命令就是用来添加、修改和删除资源群组用的。

16．esxcfg-scsidevs

- 功能：给出 ESX 上的 SCSI 设备。

```
[root@esxu2 /]# esxcfg-scsidevs
esxcfg-scsidevs <options>
Print the mappings between vmhba names and /dev names
-l|--list                 List all Logical Devices known on this
                          system with device information.
-c|--compact-list          List all Logical Devices each on a
                          single line, with limited information.
-u|--uids                 List all Device Unique Identifiers with their primary name.
-d|--device               Used to filter the --list, --compact-list
                          and uids commands to limit output to a
                          specific device.
-m|--vmfs                 Print mappings for VMFS volumes to their
                          Service Console partitions and vmhba names.
-f|--vfat                 Print mappings for VFAT volumes to their
                          Service Console partitions and vmhba names.
-a|--hbas                 Print HBA devices with identifying information
-A|--hba-device-list      Print a mapping between HBAs and the devices it
                          provides paths to.
-o|--offline-cos-dev      Offline the COS device corresponding to this vmkernel
                          device.
-n|--online-cos-dev       Bring online the COS device corresponding to this vmkernel
                          device.
-h|--help                 Show this message.
[root@esxu2 /]#
```

▲ esxcfg-scsidevs

这个命令给出 ESX 主机上的 HBA 卡和 SCSI 设备之间的关系。不管是本地或是外部存储设备，都可以给出，你也可以给出补丁或这个设备的 UID。有时候你也可以用这个命令将 SCSI 的设备挂载上或是卸下，通常要使用任何 SCSI 的设备时，这个命令都可以完全办到。

17．esxcfg-swiscsi

- 功能：vSphere 支持硬件和软件的 iSCSI。

```
[root@esxu2 /]# esxcfg-swiscsi
Usage: esxcfg-swiscsi options
  -e, --enable          Enable sw iscsi
  -d, --disable         Disable sw iscsi
  -q, --query           Check if sw iscsi is on/off
  -s, --scan            Scan for disk available through sw iscsi interface
  -k, --kill            Try to forcibly remove iscsi sw stack
  -r, --restore         Restore sw iscsi configuration from file
                        (FOR INTERNAL USE ONLY)
  -h, --help            Show this message
[root@esxu2 /]#
```

▲ esxcfg-swiscsi

在软件的 iSCSI 中，VMKernel 负责将 SCSI 命令绑入 IP 中，并且使用常规的网卡来提交这些报文，和 iSCSI 的目标服务器沟通。在 vSphere 中使用 vmhba32 为 iSCSI 的软件 HBA 卡，就是使用 esxcfg-swiscsi 来落实。当使用 iSCSI 时，由于 Service console 和 VMKernel 都必须要和 iSCSI 的目标连接。

这个命令一般不会单独使用，而是必须有一连串的动作来进行。包括了下面几个步骤。

▶ 进行 esxcfg-swiscsi 的步骤

（1） 在 VMKernel 的部分添加一个通信端口到 vSwtich 上，必须连接并路由到 iSCSI 的目标上。

（2）Service Console 也必须有一个 IP 对应到 iSCSI 的目标上。

（3）使用 esxcfg-firewall 来开放一个 iSCSI 的通信端口，使用 3260 这个通信端口。

（4）使用 esxcfg-swiscsi –e 来激活 iSCSI 功能。

（5）使用 vmkiscsi-tool -D -a 192.168.1.223 vmhba32 这个命令来查找该 IP 上的目标。

（6）使用 vmkiscsi-tool -T -l vmhba32 给出该 IP 上的对应。

（7）使用 esxcfg-rescan vmhba32 来找出新的 LUN。

（8）使用 vmkiscsi-tool -L -l vmhba32 来挂上新的 LUN。

（9）使用 mgmt-vmware restart 让 vSphere Client 上的设备也接轨补丁。

18．esxcfg-volume

- 功能：操作 ESX 下快照磁盘的命令。

```
root@esxu2:/
[root@esxu2 /]#
[root@esxu2 /]# esxcfg-volume
esxcfg-volume <options>
-l|--list                                 List all volumes which have been
                                          detected as snapshots/replicas.
-m|--mount <VMFS UUID|label>              Mount a snapshot/replica volume, if
                                          its original copy is not online.
-u|--umount <VMFS UUID|label>             Umount a snapshot/replica volume.
-r|--resignature <VMFS UUID|label>        Resignature a snapshot/replica volume.
-M|--persistent-mount <VMFS UUID|label>   Mount a snapshot/replica volume
                                          persistently, if its original copy is
                                          not online.
-h|--help                                 Show this message.
[root@esxu2 /]#
```

▲ esxcfg-volume

在 ESX 上，快照和克隆用磁盘是最好用的功能，而操作快照磁盘也是 ESX 中最重要，但却是最被忽略的功能。esxcfg-volume 主要就是针对快照来进行操作。包括了给出、挂载、卸除以及永久挂载等。事实上，我们在前几章提到的 VCB、VDR，在操作进行时，就是利用 esxcfg-volume

来进行挂载及卸除的。

19．esxcfg-upgrade

- 功能：将 ESX 主机从 2.0 版补丁到新版。

```
-h|--help                                  Show this message
[root@esxu2 /]# esxcfg-upgrade
esxcfg-upgrade -h --help
               -g --convert-grub
               -f --convert-fstab
               -r --upgrade-pre-vmkernel
               -o --upgrade-post-vmkernel

The -g option may only be used with the -r option.
[root@esxu2 /]#
```

▲ esxcfg-upgrade

这个命令较少用，当前补丁 ESX 主机都使用 Update Manager 来进行，很少用这个命令了。

25.1.2 常用的网络命令

在 ESX 中，网络命令是最复杂的。而在 esxcfg-命令中，共有下面几个网络命令，我们就来看看。

▶ esxcfg 的网络命令

- esxcfg-nics
- esxcfg-route
- esxcfg-vmknic
- esxcfg-vswif
- esxcfg-vswitch

1．esxcfg-nics

- 功能：用来调整 ESX 中网卡的参数。

```
root@esxu2:/
[root@esxu2 /]#
[root@esxu2 /]# esxcfg-nics
esxcfg-nics <options> [nic]
   -s|--speed <speed>       Set the speed of this NIC to one of 10/100/1000/10000.
                            Requires a NIC parameter.
   -d|--duplex <duplex>     Set the duplex of this NIC to one of 'full' or 'half'.
                            Requires a NIC parameter.
   -a|--auto                Set speed and duplexity automatically.  Requires a NIC para
meter.
   -l|--list                Print the list of NICs and their settings.
   -r|--restore             Restore the nics configured speed/duplex settings (INTERNAL
 ONLY)
   -h|--help                Display this message.
[root@esxu2 /]#
```

▲ esxcfg-nics

esxcfg-nics 用来调整视图及配置 ESX 服务器中物理网卡的速度。一般来说，有时候我们在 vSphere Client 配置网卡的参数导致网卡失败之后，可以在这里重新配置。和一般的 ESX 命令一样，最常用的参数就是–l 给出设备。

```
[root@vaemon.com]# esxcfg-nics –l
```

```
[root@esxu2 /]# esxcfg-nics -l
Name    PCI      Driver      Link Speed     Duplex MAC Address       MTU    Descriptio
n
vmnic0  02:00.00 e1000       Up   1000Mbps  Full   00:0c:29:54:d1:be 1500   Intel Corp
oration 82545EM Gigabit Ethernet Controller (Copper)
vmnic1  02:02.00 e1000       Up   1000Mbps  Full   00:0c:29:54:d1:c8 1500   Intel Corp
oration 82545EM Gigabit Ethernet Controller (Copper)
[root@esxu2 /]#
```

▲ 运行结果

2. esxcfg-route

- 功能：配置 ESX 主机中 VMKernel 通信端口的网关及路由 IP。

```
root@esxu2:/
[root@esxu2 /]#
[root@esxu2 /]# esxcfg-route
VMkernel default gateway is 192.168.1.1
[root@esxu2 /]#
```

▲ esxcfg-route

在 ESX 主机中，由于 Service Console 和 VM Port Group 是一开始就配置好的，因此大部分在网络会弹出问题地方都是 VMkernel。因此使用 esxcfg-route 这个命令可以手动配置网关等数据。在没有参数的情况下，会给出当前这台 ESX 主机上 VMKernel 的网关，如果有跟 IP 的话，就会配置这个 VMKernel 的新网关。

也可以手动新建新的路由给 VMKernel 使用，使用—a 参数，也可以使用—d 参数将路由删除，只要后面跟着 IP 位置即可。

3. esxcfg-vmknic

- 功能：配置 vSwitch 上的虚拟网卡，而非 ESX 上的物理网卡。

```
[root@esxu2 /]# esxcfg-vmknic
esxcfg-vmknic <options>
-a|--add                      Add a VMkernel NIC to the system,
                              requires IP parameters and
                              connection point(portgroup/dvs-name & dvport-id).
-d|--del                      Delete VMkernel NIC on the system
                              requires connection point(portgroup/dvs-name & dvport-id
).
-e|--enable                   Enable the given NIC if disabled.
-D|--disable                  Disable the given NIC if enabled.
-p|--portgroup                Setting portgroup as connection point.
-s|--dvs-name                 Setting DVSwitch name of the connection point.
                              This requires --dvport-id option to
                              be given in the same command.
-v|--dvport-id                Setting DVPort ID of connection point.
                              This requires --dvs-name option to
                              be given in the same command.
-l|--list                     List VMkernel NICs.
-i|--ip <IP Spec>             Where <IP Spec> can be one of the following...
                              <X.X.X.X>      - to use static IPv4 address specified
                              DHCP           - to enable IPv4 DHCP for address.
                              <X:X:X::/X>    - to use static IPv6 address
                              DHCPV6         - to enable IPv6 DHCP for address.
                              AUTOCONF       - to enable IPv6 Router advertised addres
```

▲ esxcfg-vmknic

在每一台 ESX 上都必须有一台虚拟交换机，而和虚拟交换机连接的就是虚拟网卡。虚拟网卡可以从 Service console、VM Port Group 或是 VMKernel 连接上虚拟交换机，这个命令就是用来配置虚拟交换机上的 VMKernel 的通信端口协议。

我们在前面的章节也说过，VMKernel 是一个很特殊的通信端口组，主要的功能就是用来给定 VMKernel 上的 IP，用来激活 VMotion，软件的 iSCSI 以及 NFS 的访问，以及 FT Logging 等

功能。

事实上，当我们在 ESX 中的虚拟交换机中有任何 Port Group 需要更改或是新建、删除时，都可以使用 esxcfg-vmknic 这个命令。但要注意的是，如果你新建的 VMKernel 要激活 VMotion，则必须进入 vSphere Client 配置，在这命令是无法使用的。

4．esxcfg-vswif

■ 功能：管理 ESX 中的 Service Console。

```
[root@esxu2 /]# esxcfg-vswif
esxcfg-vswif <options> [<vswif>]
-a|--add                      Add vswif, requires IP parameters.
                              Automatically enables interface.
-d|--del                      Delete vswif.
-l|--list                     List configured vswifs.
-e|--enable                   Enable this vswif interface.
-s|--disable                  Disable this vswif interface.
-p|--portgroup                Set the portgroup name of the vswif.
-V|--dvs-name                 Set DVPort's DVSwitch name of the vswif.
                              Requires --dv-port-id to be specified
-P|--dv-port-id               Set DVPort's DVPort ID of the vswif.
                              Requires --dvs-name to be specified
-i|--ip <IP Spec>             Where <IP Spec> can be one of the following...
                              <X.X.X.X>       - to use static IPv4 address specified
                              DHCP            - to enable IPv4 DHCP for address.
                              <X:X:X::/X>     - to use static IPv6 address
                              DHCPV6          - to enable IPv6 DHCP for address.
                              AUTOCONF        - to enable IPv6 Router advertised addres
s.
-U|--unset-ip <IP Spec>       Where <IP Spec> can be one of the following...
                              V4              - to unset IPv4 address.
                              <X:X:X::/X>     - to remove the specified IPv6 address
                              DHCPV6          - to disable IPv6 DHCP address.
                              AUTOCONF        - to disable IPv6 Router advertised flag.
-n|--netmask <X.X.X.X>        The IP netmask for this vswif.
-b|--broadcast <X.X.X.X>      The IP broadcast address for this vswif
                              (not required if netmask and ip are set).
```

▲ esxcfg-vswif

在 ESX 中，真正最重要的 IP 就是 Service Console 了。如果 Service Console 出问题，整个 ESX 就无法操作了。因此我们通常会创建新的 vSwitch，并且创建多个 Service Console 来保障 ESX 可以操作。在 ESX 中，Service Console 在 VMKernel 之中称为 vswif，在安装 ESX 时，一定会创建第一个 Service Console，要不然是无法总控的，笔者在此强烈建议一定要创建第二个以上的 Service Console。

就和其他的 esxcfg 命令一样，esxcfg-vswif 可以视图、创建、修改、删除 Service Console，第一个示例，我们就来看看视图当前使用的 Service Console。

▶ 视图当前的 Service Console

```
esxcfg-vswif -l
```

```
[root@esxu2 /]# esxcfg-vswif -l
Name     Port Group/DVPort    IP Family IP Address                          Netmask
                                  Broadcast       Enabled    TYPE
vswif0   Service Console      IPv4      192.168.1.213                       255.255
.255.0                            192.168.1.255   true       STATIC
[root@esxu2 /]#
```

▲ 运行结果

如果我们想要添加第二个 Service Console，就要用这个命令。当然我们在开始之前要先创建一个 Port Group。下面的例子就是创建一个新的 Service Console。

▶ 创建新的 Service Console

```
esxcfg-vswitch --add-pg="Service Console 2" vSwitch1

esxcfg-vswif -a -i 192.168.1.215 -n 255.255.0.0 -p "Service Console 2" vswif1
```

```
root@esxu2:/
[root@esxu2 /]#
[root@esxu2 /]# esxcfg-vswitch --add-pg="Service Console 2" vSwitch1
[root@esxu2 /]# esxcfg-vswif -a -i 192.168.1.215 -n 255.255.0.0 -p "Service Console 2"
 vswif1
[2010-07-11 14:49:15 'Vnic' warning] Generated New MAC address, 00:50:56:43:91:af for
vswif1
Nothing to flush.
[root@esxu2 /]#
```

▲ 运行结果

我们可以使用—l 参数来给出刚才创建的 Service Console。

5. esxcfg-vswitch

- 功能：管理 ESX 主机中的虚拟交换机。

```
[root@esxu2 /]# esxcfg-vswitch
esxcfg-vswitch [options] [vswitch[:ports]]
  -a|--add                    Add a new virtual switch.
  -d|--delete                 Delete the virtual switch.
  -l|--list                   List all the virtual switches.
  -L|--link=pnic              Set pnic as an uplink for the vswitch.
  -U|--unlink=pnic            Remove pnic from the uplinks for the vswitch.
  -R|--restore-uplinks        Restore uplinks for all vswitches from config file.
                              (FOR INTERNAL USE ONLY).
  -M|--add-pg-uplink=uplink   Add an uplink to the list of uplinks for a portgroup
  -N|--del-pg-uplink=uplink   Delete an uplink from the list of uplinks for a portgrou
p
  -P|--add-dvp-uplink=uplink  Add an uplink to a DVPort on a DVSwitch.
                              Must specify DVPort Id.
  -Q|--del-dvp-uplink=uplink  Delete an uplink from a  DVPort on a DVSwitch.
                              Must specify DVPort Id.
  -V|--dvp=dvport             Specify a DVPort Id for the operation.
  -p|--pg=portgroup           Specify a portgroup for operation
                              Use ALL to set VLAN IDs on all portgroups
  -v|--vlan=id                Set vlan id for portgroup specified by -p
                              0 would disable the vlan
  -c|--check                  Check to see if a virtual switch exists.
                              Program outputs a 1 if it exists, 0 otherwise.
```

▲ esxcfg-vswitch

这个命令是所有 ESX 命令中最重要的一个。让你可以完全管理 ESX 主机中的虚拟交换机。大家都知道虚拟交换机是 ESX 中的网络基本部件，因此如果需要通过命令来管理虚拟交换机，只能用这个命令。esxcfg-vswitch 最简单的参数和其他命令一样，就是给出当前拥有的虚拟交换机—l。

▶ **给出当前 ESX 的虚拟交换机**

```
esxcfg-vswitch -l
```

```
[root@esxu2 /]# esxcfg-vswitch -l
Switch Name    Num Ports   Used Ports  Configured Ports  MTU     Uplinks
vSwitch0       64          5           64                1500    vmnic0

  PortGroup Name        VLAN ID  Used Ports  Uplinks
  VM Network            0        1           vmnic0
  Service Console       0        1           vmnic0
  VMotion               0        1           vmnic0

Switch Name    Num Ports   Used Ports  Configured Ports  MTU     Uplinks
vSwitch1       64          2           64                1500

  PortGroup Name        VLAN ID  Used Ports  Uplinks
  Test                  0        0
  Service Console 2     0        1

[root@esxu2 /]#
```

▲ 运行结果

如果你在升级 ESX 之后发现 Service Console 不见了，这个命令就是你最好的朋友。一般来说，我们使用这个命令时，会先给出所有的虚拟交换机，使用—l 参数。如果你想新建虚拟交换机，则使用—a 来新建交换机。我们就举一个例子来新建新的虚拟交换机。

▶ 新建虚拟交换机

```
esxcfg-vswitch -a vSwitch1

esxcfg-vswitch -l
```

```
[root@esxu2 /]# esxcfg-vswitch -a vSwitch2
[root@esxu2 /]# esxcfg-vswitch -l
Switch Name    Num Ports   Used Ports  Configured Ports  MTU     Uplinks
vSwitch0       64          5           64                1500    vmnic0

  PortGroup Name       VLAN ID  Used Ports  Uplinks
  VM Network           0        1           vmnic0
  Service Console      0        1           vmnic0
  VMotion              0        1           vmnic0

Switch Name    Num Ports   Used Ports  Configured Ports  MTU     Uplinks
vSwitch1       64          2           64                1500

  PortGroup Name       VLAN ID  Used Ports  Uplinks
  Test                 0        0
  Service Console 2    0        1

Switch Name    Num Ports   Used Ports  Configured Ports  MTU     Uplinks
vSwitch2       64          1           64                1500

  PortGroup Name       VLAN ID  Used Ports  Uplinks
```

▲ 运行结果

我们可以新建一个新的虚拟交换机。但要注意的是，这个新建的虚拟交换机并没有任何 Service Console 或是 VMKernel，因此不会在 vSphere Client 中显示。但是在命令中已经可以看到了。你也可以使用--add 来新建，和-a 是完全一样的命令。

在创建了新的虚拟交换机之后，就要加入 Port Group 了。我们可以键入-A 来新建 Port Group。

```
esxcfg-vswitch -A "Test "  vSwitch1

esxcfg-vswitch -l
```

```
[root@esxu2 /]# esxcfg-vswitch -A "Test1" vSwitch1
[root@esxu2 /]# esxcfg-vswitch -l
Switch Name    Num Ports   Used Ports  Configured Ports  MTU     Uplinks
vSwitch0       64          5           64                1500    vmnic0

  PortGroup Name       VLAN ID  Used Ports  Uplinks
  VM Network           0        1           vmnic0
  Service Console      0        1           vmnic0
  VMotion              0        1           vmnic0

Switch Name    Num Ports   Used Ports  Configured Ports  MTU     Uplinks
vSwitch1       64          2           64                1500

  PortGroup Name       VLAN ID  Used Ports  Uplinks
  Test1                0        0
  Test                 0        0
  Service Console 2    0        1

Switch Name    Num Ports   Used Ports  Configured Ports  MTU     Uplinks
vSwitch2       64          1           64                1500

  PortGroup Name       VLAN ID  Used Ports  Uplinks

[root@esxu2 /]#
```

▲ 运行结果

我们也可以使用--add-pg 来新建 Port Group。当我们创建好 vSwitch 之后，事实上在 vSphere Client 上是无法看到的。因为我们还没有将这个虚拟交换机连接到网卡上。因此接下来我们必须将这个交换机连接到 vmnic 上。而两者之间的连接在 ESX 中称为 Uplink。在连接之后，这个虚拟交换机才能占有这片网卡，而整个虚拟交换机才能常规使用。此时我们必须使用—L 命令来连接虚拟交换机和物理网卡。

▶ 连接虚拟交换机和物理网卡

```
esxcfg-vswitch -L vmnic1 vSwitch1
```

```
root@esxu2:/
[root@esxu2 /]#
[root@esxu2 /]# esxcfg-vswitch -L vmnic1 vSwitch1
[root@esxu2 /]#
```

▲ 运行结果

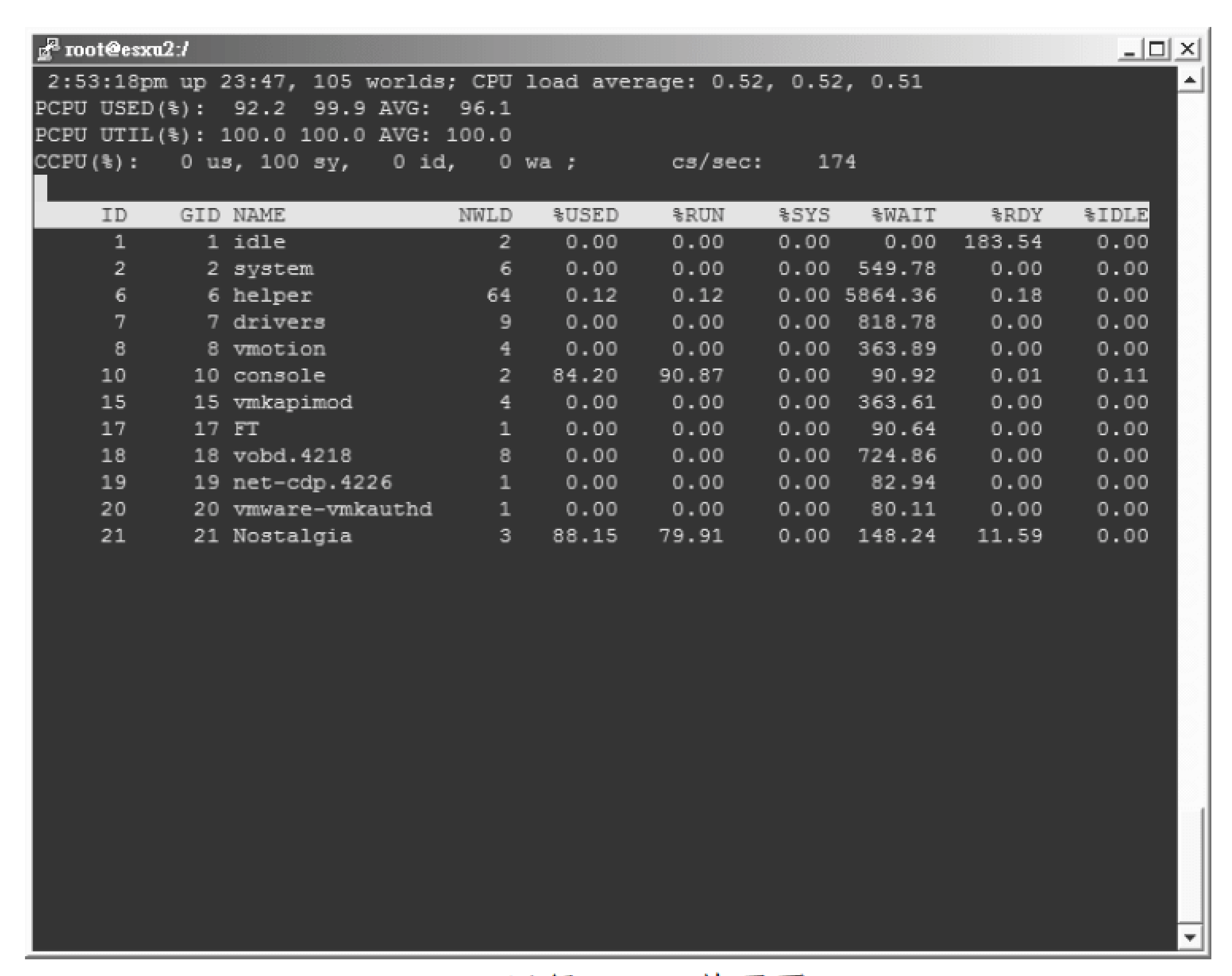
以上的命令将会把 vmnic1 连接上 vSwitch1，如果连接失败的话，可以用-U 命令撤销连接。

▶ --esx 的其他命令

在 ESX 主机还有其他的三个命令，但我们通常常用的只有一个 esxtop，这里简单介绍一下。

▶ esx 命令

- Esxcli：用来管理 VMware Pluggable Storage Array（PSA）的命令，较少用。
- Esxtop：等于是 Linux 下的 top 命令，可以监视 ESX 主机的硬件状态，很常用。
- Esxupdate：补丁 ESX 主机，但补丁 ESX 是一个大工程，我们会尽量用 Update Manager。在使用这个命令时，所有的动作或补丁表项都会留在/var/log/vmware/esxupdate.log 中。



▲ 运行 esxtop 的画面

25.2 其他的 ESX 命令

除了标准的 esxcfg 命令之外，ESX 上当然还有许多辅助的命令，我们在这一小节就来看看这些命令。

25.2.1 ESX 所附的命令

ESX 所附的命令包括几大类，主要还是针对 ESX 主机本身的功能，而非 vSphere 的功能。举例来说，处理 VMDK 磁盘格式的 vmkfstools 就是用来处理 ESX 上的磁盘格式，这一类命令通常都针对 ESX 的 Linux 功能，下面我们将一一介绍。

1. vmkfstools

- 功能：处理 ESX 所管理的虚拟硬盘。

```
[root@esxu2 /]# vmkfstools
No valid command specified


OPTIONS FOR FILE SYSTEMS:

vmkfstools -C --createfs vmfs3
               -b --blocksize #[mMkK]
               -S --setfsname fsName
           -Z --spanfs span-partition
           -G --growfs grown-partition
   deviceName

           -P --queryfs -h --humanreadable
           -T --upgradevmfs
               -x --upgradetype [zeroedthick|eagerzeroedthick|thin]
           -u --upgradefinish
   vmfsPath

OPTIONS FOR VIRTUAL DISKS:

vmkfstools -c --createvirtualdisk #[gGmMkK]
               -d --diskformat [zeroedthick|
```

▲ vmkfstools

这个命令是处理 ESX 所管理的所有虚拟硬盘最直白的手段，从前使用这个命令来转换不同格式的 VMDK，但现在不但可以用来转换，更可以添加容量、压缩、管理所有的 VMDK。可以说，所有 ESX 下有关虚拟硬盘的管理，都是使用这个命令。

我们可以使用这个命令来创建 VMFS 的文件系统。举例来说，如果我们要将某一个 HBA 卡上的 LUN 创建一个 VMFS 的分割区，就可以键入：

▶ 在 LUN 上创建一个 VMFS 的分割区

```
vmkfstools -C vmfs3 -b 2m -S datastore01 vmhba1:0:1:1
```

其中—C 是创建 VMFS 的分割区，2m 是创建 2MB 的 Block Size，datastore01 是这个 Datastore 的名称，vmhba1 是第一片 HBA 卡，0 是第一个分割区，1 是第一个 LUN。

由于 vSphere 的 Datastore 是可以不只一个 LUN 所组成，因此我们可以用这个命令来落实 Datastore 的扩展，但笔者还是建议使用 vSphere Client 界面来落实。

除了处理 VMDK 的创建之外，也可以处理已经创建好的虚拟磁盘。举例来说—X 参数就可以将原来的 VMDK 扩展容量。下面的例子是将一个原来为 100GB 的 VMDK 磁盘扩展成 200GB。要注意的是这个命令必须键入“新的大小”，而非你想要扩展的大小，因此必须注意新的大小要大于原来的大小。

▶ 扩展虚拟磁盘的容量

```
vmkfstools -X 20g /vmfs/volumes/storage1/vmtest.vmdk
```

```
[root@esxu2 /]#
[root@esxu2 /]# vmkfstools -X 20g /vmfs/volumes/Storage1/Nostalgia/Nostalgia.vmdk
[root@esxu2 /]#
```

▲ 运行结果，需要较长的时间

一般来说，我们可以克隆一个 VMDK 文件，克隆的用意就是将这个 VMDK 给另一个虚拟机来使用，而不是单纯的 COPY 而已。

▶ 克隆 VMDK

```
vmkfstools -i /vmfs/volumes/Storage1/2003/2003.vmdk -d thin /vmfs/volumes/
Storage1/2003c/2003c.vmdk
```

```
[root@esxu2 /]# vmkfstools -i /vmfs/volumes/Storage1/Nostalgia/Nostalgia.vmdk /vmfs/vo
lumes/Storage1/n.vmdk
Destination disk format: VMFS zeroedthick
Cloning disk '/vmfs/volumes/Storage1/Nostalgia/Nostalgia.vmdk'...
Failed to clone disk : There is not enough space on the file system for the selected o
peration (13).
[root@esxu2 /]#
```

▲ 克隆 VMDK

上面的命令使用—i 来进行虚拟机 VMDK 的克隆，来源是/vmfs/volumes/Storage1/2003/2003.vmdk，目的是/vmfs/volumes/Storage1/2003c/2003c.vmdk，特别要注意的是其文件格式为 thin，意味着这个转换出来的文件是 Thin Provisioning 的，也就是文件的大小会随着使用而成长，而非一次就固化大小。这个功能是在 vSphere Client 中无法达成的。

如果你在创建 VMDK 时，使用的是 Thin Provisioning，但想将这个 VMDK 转换回常规的 Thick Provisioning，可以键入下面的命令。

▶ 转换 VMDK 的格式

```
vmkfstools -j /vmfs/volumes/Storage1/2003/2003.vmdk
```

```
[root@esxu2 /]#
[root@esxu2 /]# vmkfstools -j /vmfs/volumes/Storage1/Nostalgia/Nostalgia.vmdk
The virtual disk (/vmfs/volumes/Storage1/Nostalgia/Nostalgia.vmdk) cannot be inflated
type: <11>, capacity: <21474836480>, space used: <21474836480>.
[root@esxu2 /]#
```

▲ 转换 VMDK

当我们使用 Thin Provisioning 来规划硬盘时，也许有时因为效能的问题而需要将这个硬盘转换成真实对应而非自动增长大小的形态，此时可以用—j 命令还原，这在 vSphere Client 中也做不到的。

2．vmware-cmd

▪ 功能：用来激活或是配置 VM，也可以给出当前的 VM 总数。

```
root@esxu2:/
    -P <password>    specifies a password

  Server Operations:
    /usr/bin/vmware-cmd -l
    /usr/bin/vmware-cmd -s listvms
    /usr/bin/vmware-cmd -s register <config_file_path>
    /usr/bin/vmware-cmd -s unregister <config_file_path>

  VM Operations:
    /usr/bin/vmware-cmd <cfg> getstate
    /usr/bin/vmware-cmd <cfg> start [soft|hard|trysoft]
    /usr/bin/vmware-cmd <cfg> stop [soft|hard|trysoft]
    /usr/bin/vmware-cmd <cfg> reset [soft|hard|trysoft]
    /usr/bin/vmware-cmd <cfg> suspend [soft|hard|trysoft]
    /usr/bin/vmware-cmd <cfg> getconfig <variable>
    /usr/bin/vmware-cmd <cfg> setguestinfo <variable> <value>
    /usr/bin/vmware-cmd <cfg> getguestinfo <variable>
    /usr/bin/vmware-cmd <cfg> getproductinfo <product_info>
    /usr/bin/vmware-cmd <cfg> connectdevice <device_name>
    /usr/bin/vmware-cmd <cfg> disconnectdevice <device_name>
    /usr/bin/vmware-cmd <cfg> getid
    /usr/bin/vmware-cmd <cfg> getconfigfile
    /usr/bin/vmware-cmd <cfg> getheartbeat
    /usr/bin/vmware-cmd <cfg> getuptime
    /usr/bin/vmware-cmd <cfg> gettoolslastactive
    /usr/bin/vmware-cmd <cfg> hassnapshot
    /usr/bin/vmware-cmd <cfg> createsnapshot <name> <description> <quiesce> <memory>
    /usr/bin/vmware-cmd <cfg> revertsnapshot
    /usr/bin/vmware-cmd <cfg> removesnapshots
    /usr/bin/vmware-cmd <cfg> answer

Invalid operation specified.
[root@esxu2 /]#
```

▲ vmware-cmd

vmware-cmd 是一个万用的命令，可以处理服务器，也可以处理 VM。在处理服务器上面，就是给出其上的 VM，并且也可以配置 VM 的参数。举例来说，如果要将一个 VM 加入，并且引导，就要键入下面的命令。

▶ 加入 VM 并激活

```
vmware-cmd -s register /vmfs/volumes/VMs/vm1/vm1.vmx
vmware-cmd /vmfs/volumes/VMs/vm1/vm1.vmx start
```

```
root@esxu2:/
[root@esxu2 /]#
[root@esxu2 /]# vmware-cmd -s register /vmfs/volumes/Storage1/Nostalgia/Nostalgia.vmx
vim.fault.AlreadyExists: The specified key, name, or identifier already exists.
[root@esxu2 /]#
[root@esxu2 /]# vmware-cmd /vmfs/volumes/Storage1/Nostalgia/Nostalgia.vmx start
start() = 1
[root@esxu2 /]#
```

▲ 运行结果

如果你希望一个 VM 能立即关机，并且完全不等待任何回馈，可以键入。

▶ 让 VM 立即关机

```
vmware-cmd /vmfs/volumes/VMs/vm1/vm1.vmx stop hard
```

```
start() = 1
[root@esxu2 /]# vmware-cmd /vmfs/volumes/Storage1/Nostalgia/Nostalgia.vmx stop hard
stop(hard) = 1
[root@esxu2 /]#
```

▲ 运行结果

这个 VM 就会立即关机，速度会非常地快。

3. vm-support

- 功能：将 ESX 主机上的各种配置打包并且提交到 VMware。

```
root@esxu2:/
[root@esxu2 /]#
[root@esxu2 /]# vm-support

VMware ESX Support Script 1.30

Preparing files: |
```

▲ vm-support

如果你的 ESX 主机出问题了，可以使用这个命令将所有的 ESX 配置文件打包成 TAR 格式并且提交回 VMware 的网站。当然你必须是有购买 ESX 主机授权的用户才行。一般会使用—x 参数，将不需要加入的目录排除。一般来说，我们也可以使用这个命令来给出性能参数让 esxtop 来可读。举例来说，我们想每 30 秒监视这个主机，一共持续 10 分钟，可以键入：

▶ **监视 ESX 的性能**

```
Vm-support -S -i 30 -d 600
```

```
See man page for more information on vm-support version 1.30
[root@esxu2 /]# vm-support -S -d 600 -i 30

VMware ESX Support Script 1.30


Taking performance snapshots.  This will take about 600 seconds.

Starting detailed scheduler stats.

Starting vscsiStats.
```

▲ 运行结果

4. vimsh

- 功能：提供一个特殊的 Shell 环境给 ESX 的主机使用。

```
root@esxu2:/
[root@esxu2 /]#
[root@esxu2 /]# vimsh
Using system libcrypto, version 9080CF

[2010-07-11 15:24:48.194 F667E6D0 info 'App'] Current working directory: /
[2010-07-11 15:24:48.197 F667E6D0 info 'App'] Vmacore::InitSSL: doVersionCheck = true,
 handshakeTimeoutUs = 20000000
[2010-07-11 15:24:48.256 F64FBB90 warning 'Libs'] [Vmdb_Unset] Unsetting unknown path:
 /vmomi/
[/]$
```

▲ vimsh

vimsh 是 ESX 中最重要的功能之一，它提供了一个 Shell，让用户可以在其中进行 Script 文件的编写。整个 vimsh 的功能使用就可以写一本 500 页的书了。如果你想要开发全自动的 ESX 程序，一定要用 vimsh 来进行。不过当前市面上还是很少有这一方面的数据。

5. vmware-vim-cmd

- 功能：提供直接在 Service Console 下运行草稿档命令，不需要留在 vimsh 中。

```
root@esxu2:/
[root@esxu2 /]#
[root@esxu2 /]# vmware-vim-cmd
Commands available under /:
hostsvc/          proxysvc/          supportsvc_cmds/  vmsvc/
internalsvc/      solo/              vimsvc/           help
[root@esxu2 /]#
```

▲ vmware-vim-cmd

我们使用 vimsh 时，在运行完命令后会留在 vimsh 中。但我们可以使用 vmware-vim-cmd 的话，可以直接在 Service Console 中运行 vimsh 的命令，并且速度也更快。

6．vmkload_mod

- 功能：直接装载或卸下 ESX 中模块的工具。

```
root@esxu2:/
[root@esxu2 /]#
[root@esxu2 /]# vmkload_mod
Missing module name.
Usage: vmkload_mod <options> <module> <param_name=value ...>
Options:
   -b --brief
   -e --exportsym
   -h --help
   -l --list
   -m --mod-name <name>
   -s --showinfo
   -u --unload
   -v --verbose <level>
[root@esxu2 /]#
```

▲ vmkload_mod

在引导时，ESX 会固化装载一些模块，但有时你需要在 Service Console 挂载或卸下模块时，在不需要引导的情况下，使用 vmkload_mod 即可。

7．vmkping

- 功能：使用 VMKernel 来 ping 其他的 ESX 主机。

```
root@esxu2:/
[root@esxu2 /]#
[root@esxu2 /]# vmkping 192.168.1.1
PING 192.168.1.1 (192.168.1.1): 56 data bytes
64 bytes from 192.168.1.1: icmp_seq=0 ttl=64 time=1.943 ms
64 bytes from 192.168.1.1: icmp_seq=1 ttl=64 time=0.915 ms
64 bytes from 192.168.1.1: icmp_seq=2 ttl=64 time=0.871 ms

--- 192.168.1.1 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 0.871/1.243/1.943 ms
[root@esxu2 /]#
```

▲ vmkping

vmkping 和 ping 不一样的地方，就是其不使用 Linux 的 IP Stack，而是使用 VMKernel 的 IP Stack 来 ping 主机。主要是用来嗅探 VMotion、FT、NAS、iSCSI 这些设备的连通性。vmkping 使用自身本身的通信端口来 ping。如果使用了参数—D，则会自动 ping 各个重要的点，如 iSCSI 、默认的网关等。

25.2.2 在 ESX 下的配置文件

我们在操作 ESX 下的各种命令或是 vSphere Client 的动作时，事实上就是补丁一些配置文件。由于 Linux 下是以文本形态的配置文件为主，因此我们可以手动更改这些配置文件也可以达到一

样的目的，在这一小节，我们就来看看这些配置文件。

1. /etc/vmware/esx.conf

这是所有 ESX 的配置文件，取代了旧版 VMware 的多重配置。如果你有多个 ESX 主机，并且配置完全一样，就可以将这个配置文件复制到其他的主机，是 ESX 中最重要的一个配置文件。这些配置文件的条目都很直白，读者们可以直接使用 vi 进入检视。

```
root@esxu2:/
/adv/Migrate/Vmknic = "vmk0"
/adv/Misc/CosCorefile = "/vmfs/volumes/4c178a4c-64d9d625-3d89-000c2905f099/esxconsole-4c17fa32-5a70-8b7d-f4f4-000c2905f099/core-dumps/cos-core"
/adv/Misc/HostIPAddr = "192.168.1.213"
/adv/Misc/HostName = "esxu2"
/adv/Net/ManagementAddr = "192.168.1.213"
/adv/Net/ManagementIface = "vswif0"
/adv/UserMem/UserMemASRandomSeed = "568538408"
/advUserOptions/options[0000]/default = "1"
/advUserOptions/options[0000]/description = "Enable or Disable the CIM service"
/advUserOptions/options[0000]/hidden = "false"
/advUserOptions/options[0000]/max = "1"
/advUserOptions/options[0000]/min = "0"
/advUserOptions/options[0000]/name = "CIMEnabled"
/advUserOptions/options[0000]/type = "int"
/advUserOptions/options[0001]/default = "1"
/advUserOptions/options[0001]/description = "Enable or Disable the CIM OEM Providers"
/advUserOptions/options[0001]/hidden = "false"
/advUserOptions/options[0001]/max = "1"
/advUserOptions/options[0001]/min = "0"
/advUserOptions/options[0001]/name = "CIMOEMProvidersEnabled"
/advUserOptions/options[0001]/type = "int"
/advUserOptions/options[0002]/default = "1"
/advUserOptions/options[0002]/description = "Enable or Disable the CIM Custom Providers"
/advUserOptions/options[0002]/hidden = "false"
/advUserOptions/options[0002]/max = "1"
/advUserOptions/options[0002]/min = "0"
/advUserOptions/options[0002]/name = "CIMCustomProvidersEnabled"
/advUserOptions/options[0002]/type = "int"
/advUserOptions/options[0003]/default = "60"
@
"/etc/vmware/esx.conf" 478L, 33007C
```

▲ 配置文件内容

2. /etc/nsswitch.conf

这是 ESX 主机的主机名的配置文件。如果你的 ESX 主机有多种名称，并且需要有固化的顺序时，就需要更改这个文件。这个选项最常用的场合就是远程总控（通过互联网）。这个文件的参数有很多，但要配置的通常是以 hosts 开头的。举例来说：

▶ hosts: files nisplus dns

这个文件就会以/etc/hosts 为主可读，接下来会可读 NIS+，接下来会以 DNS 给定的名称为主。DNS 的配置文件则是存放在/etc/resolv.conf file。由于这个文件也很直白，因此读者也可以直接用 vi 来检视。

```
root@esxu2:/
# Autogenerated by esxcfg-auth

aliases:        files nisplus
automount:      files nisplus
bootparams:     nisplus [NOTFOUND=return] files
ethers:         files
group:          files
hosts:          files dns
netgroup:       nisplus
netmasks:       files
networks:       files
passwd:         files
protocols:      files
publickey:      nisplus
rpc:            files
services:       files
shadow:         files
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
"/etc/nsswitch.conf" 17L, 369C
```

▲ 配置文件内容

3. /usr/bin/vmware-watchdog

这是一个系统监视的文件，用来监视 hostd 是否失效，如果失效就会重新激活。

4. hostd

这是激活 ESX 主机功能的重要服务，主要是负责 ESX 上的管理功能。通常我们在运行 vSphere Client 时，如果弹出了 Host not responding 的错误消息时，主要就是 hostd 没有激活，此时如果将 hostd 激活，vSphere Client 就可以快速登录了。要激活 hostd 最快的手段就是键入下列命令：

▶ Service mgmt-vmware restart

```
root@esxu2:/
[root@esxu2 /]#
[root@esxu2 /]# service mgmt-vmware restart
Stopping VMware ESX Management services:
   VMware ESX Host Agent Watchdog                       [  OK  ]
   VMware ESX Host Agent                                [  OK  ]
Starting VMware ESX Management services:
   VMware ESX Host Agent (background)                   [  OK  ]
   Availability report startup (background)             [  OK  ]
[root@esxu2 /]#
```

▲ 运行结果

Hostd 的配置文件则是存放在/var/log/vmware/hostd.log 中。

5. /etc/vmware/firewall/services.xml

前面提到 esxcfg-firewall 这个程序，主要开放的服务就是在这个文件中定义。这些定义都会有名称和通信端口的号码。你当然可以自身在此中定义自身的服务，但是 VMware 不建议这么做。前面提到的 hostd 在激活时会装载这个文件。

```
root@esxu2:/
<!-- Firewall configuration information -->
<ConfigRoot>

  <!-- Known and blessed services -->

  <service id='0000'>
    <id>sshServer</id>
    <rule id='0000'>
      <direction>inbound</direction>
      <protocol>tcp</protocol>
      <port type='dst'>22</port>
      <flags>-m state --state NEW</flags>
    </rule>
  </service>

  <service id='0001'>
    <id>sshClient</id>
    <rule>
      <direction>outbound</direction>
      <protocol>tcp</protocol>
      <port type='dst'>22</port>
      <flags>-m state --state NEW</flags>
    </rule>
  </service>

  <service id='0002'>
    <id>ftpServer</id>
    <rule>
      <direction>inbound</direction>
      <protocol>tcp</protocol>
      <port type='dst'>21</port>
      <flags>-m state --state NEW</flags>
"/etc/vmware/firewall/services.xml" 390L, 9532C
```

▲ 配置文件内容

25.2.3 VCB 使用的脚本文件

我们在前面的章节说明了安装 VCB 之后，Backup 主机就会拥有许多 VCB 的运行程序，但是在 ESX 本身也会安装 VCB 的命令来针对其上的 VM 运行，我们就来看看。

1. vcbVMName

这和 Windows 2003 下的 vcbVMName 的功能没差别，就是激活 VCB 的副本机制，读者可以引用前面章节的语法参数。

```
root@esxu2:/
[root@esxu2 /]#
[root@esxu2 /]# vcbVmName
[2010-07-11 15:29:11.318 F4F39B60 info 'App'] Current working directory: /
[2010-07-11 15:29:11.810 F4F39B60 error 'vcbVmName'] Error: Missing command line optio
n.
VMware Consolidated Backup -- VM Locator Utility
Version 4.0.0 (build-261974)
Copyright (C) 1998-2010 VMware, Inc.  All rights reserved.
Usage:
vcbVmName -h <url> -u <username> [-p <password>]
          -s <searchSpec> [-c <cachefile>] [-e (0|1)] [<verbosity>]

      <url>    := <hostname>[":"<port>]
 <username>    := <string>
 <password>    := <string>
 <searchSpec> := <Type>":"<Qualifier>
     <Type>    := "Any"|"IpAddr"|"MoRef"|"Uuid"|"Name"|"Powerstate"
<Qualifier>    := <string>
<cachefile>    := <filename>
<extended info> := -e (0|1)
<verbosity>    := -L (0-6)

If no password is specified on the command line, you will be prompted for one.

Example:
Use the following command line to locate a VM with the host name
"buvm01.eng.vmware.com" on the ESX server "esx17.vmware.com"
using the user name "sdk" and the password "foobar":

vcbVmName -h esx17.vmware.com -u sdk -p foobar -s ipaddr:buvm01.eng.vmware.com

Consult the VI3 Backup Guide for more information.
[root@esxu2 /]# ~
```

▲ vcbVMName

2. vcbSnapshot

这也是 Windows 2003 下有的命令。事实上在 ESX 上的 VCB 命令和 Windows 2003 下 VCB 的命令参数完全兼容。

```
root@esxu2:~
login as: root
root@192.168.1.213's password:
Last login: Sun Jul 11 12:03:13 2010 from 192.168.1.100
[root@esxu2 ~]# clear
[root@esxu2 ~]#
[root@esxu2 ~]# vcbSnapshot
[2010-07-11 16:39:14.682 F4EE2B60 info 'App'] Current working directory: /root
[2010-07-11 16:39:14.725 F4EE2B60 error 'vcbSnapshot'] Error: No operation speci
fied (one of the options 'c', 'd', 'f', 'l' must be specified).
VMware Consolidated Backup -- Snapshot Utility
Version 4.0.0 (build-261974)
Copyright (C) 1998-2010 VMware, Inc.  All rights reserved.
Usage:
vcbSnapshot -h <url> -u <username> [-p <password>] [<verbosity>] <command>
     <url>      := <hostname>[":"<port>]
     <command> := <create>|<find>|<list>|<delete>
   <verbosity> := -L (0-6)
     <create>  := -c <VM> <SnapshotName> [<Description>]
     <find>    := -f <VM> <SnapshotName>
     <list>    := -l <VM> <SS>
     <delete>  := -d <VM> <SS>
     <VM>      := <moref>|<uuid>|<ipaddr>|<...>
                  (See VM Backup guide for complete list of search
                  criteria.)
```

▲ vcbSnapshot

3. vcbSnapAll

这是将当前运行的所有 VM 做成快照的命令，这是 vSphere 中新建的命令。

```
[root@esxu2 ~]# vcbSnapAll
[2010-07-11 16:39:44.741 F4ED6B60 info 'App'] Current working directory: /root
[2010-07-11 16:39:44.769 F4ED6B60 error 'vcbSnapAll'] Error: Missing command lin
e option
VMware Consolidated Backup -- Virtual Machine Mount Utility
Version 4.0.0 (build-261974)
Copyright (C) 1998-2010 VMware, Inc.  All rights reserved.
Usage:
vcbSnapAll -h <url> -u <username> -p <password> <operation> [<verbosity>]
            <url> := <hostname>[":"<port>]
      <operation> := <mount_op>
      <verbosity> := -L (0-6)
       <mount_op> := -a <VM> <mountPoint> [<export-flags>][<datastores>]

             <VM> := <any><powerstate><...>
                     (See VM Backup Guide for complete list of search
                     criteria.)
     <mountPoint> := -r <local_dir>
     <datastores> := -C <datastore_catalog_file>
   <export-flags> := -M (0|1) -F (0|1)

Export Flags:
 -M: If set, the disk is exported into a single (monolithic) file.
     When turned off (default), the disk is split into multiple 2GB files.
```

▲ vsbSnapAll

4. vcbMounter

如果你要针对已经引导的 VM 做副本，一定要先做一个快照，使用这个命令是最方便的。

```
 is unset - Destination directory is unset.
Script was invoked with illegal or missing arguments.
[root@esxu2 ~]# clear
[root@esxu2 ~]#
[root@esxu2 ~]# vcbMounter
[2010-07-11 16:40:17.735 F4EF0B60 info 'App'] Current working directory: /root
[2010-07-11 16:40:17.755 F4EF0B60 error 'vcbMounter'] Error: Missing command lin
e option
VMware Consolidated Backup -- Virtual Machine Mount Utility
Version 4.0.0 (build-261974)
Copyright (C) 1998-2010 VMware, Inc.  All rights reserved.
Usage:
vcbMounter -h <url> -u <username> [-p <password>] <operation> [<verbosity>] [<mo
de>]
            <url> := <hostname>[":"<port>]
      <operation> := <mount_op> | <file_umount_op> | <auto_umount_op>
      <verbosity> := -L (0-6)
           <mode> := -m "cos"|"nbd"|"nbdssl"
       <mount_op> := -a <VM> <mountPoint> [<flavor>][<datastores>]
 <file_umount_op> := -d <VM> <SnapId> <mountPoint>
 <auto_umount_op> := -U <mountpoint_dir>

             <VM> := <moref>|<uuid>|<ipaddr>|<...> [-c <cacheFile>]
                     (See VM Backup guide for complete list of search
```

▲ vcbMounter

5. vcbExport

这是用来将 Snapshot 导出的程序。在 ESX 上运行是最方便的。

```
[root@esxu2 ~]# vcbExport
[2010-07-11 16:41:17.214 F4F0EB60 info 'App'] Current working directory: /root
[2010-07-11 16:41:17.269 F4F0EB60 error 'vcbExport'] Error: Missing source file/
disk specifier argument.
VMware Consolidated Backup -- Disk export utility
Version 4.0.0 (build-261974)
Copyright (C) 1998-2010 VMware, Inc.  All rights reserved.
Usage:
vcbExport [-M (0|1)] [-F (0|1)] -d <dest_file> -s <diskspec> [<verbosity>]
       <dest_file> := <local_filename>
        <diskspec> := <local_filename>
       <verbosity> := -L (0-6)

Export Disk Flags:
 -M: If set, the disk is exported into a single (monolithic) file.
     When turned off (default), the disk is split into multiple 2GB files.
 -F: If set, the disk is exported as "flat" disk, with no optimizations.
     When turned off (default), the exported disk files will be more compact as
     unused space in the disk image is not included in the  exported file.

Example:
Export a virtual disk on a locally accessible file system:
  vcbExport -s /vmfs/volumes/storage1/myvm/myvm.vmdk -d /dest/myvym.vmdk
```

▲ vcbExport

6. vcbRestore

这是用来将 Snapshot 还原的命令，在 VCB 运行之后，使用这个命令可以将 Snapshot 还原。

```
root@esxu2:~
[root@esxu2 ~]# vcbRestore
[2010-07-11 16:42:01.637 F4F30B60 info 'App'] Current working directory: /root
[2010-07-11 16:42:01.668 F4F30B60 error 'vcbRestore'] Error: Missing command lin
e option
VMware Consolidated Backup -- Virtual Machine restore utility
Version 4.0.0 (build-261974)
Copyright (C) 1998-2010 VMware, Inc.  All rights reserved.
Usage:
vcbRestore -h <url> -u <username> [-p <password>] [-C <datastorecatalog>]
           -s <sourcedir> [-a <catalog>] [-V(0|1)] [-M(0|1)] [-F(0|1)]
           [-b <behavior>] [-o <powerop>] [<verbosity>]

              <url> := <hostname>[":"<port>]
 <datastorecatalog> := <filename>
        <sourcedir> := <dirname>
          <catalog> := <filename>
         <behavior> := overwrite|prompt|keep|abort
          <powerop> := on|allon|alloff
        <verbosity> := -L (0-6)

If no password is specified on the command line, you will be prompted for one.

Import Disk Flags:
 -V: If set to 1, the disk is imported into a VMFS-style disk. This will
```

▲ vcbRestore

7. vcbResAll

这是将所有的副本还原的命令，使用的参数也很直观，读者们可以参阅命令行来运行。

```
root@esxu2:~
[root@esxu2 ~]# vcbResAll
Missing or illegal arguments: Source directory is unset.
[2010-07-11 16:42:29.903 F4F04B60 info 'App'] Current working directory: /root
[2010-07-11 16:42:29.925 F4F04B60 error 'vcbResAll'] Error: Missing command line
 option
VMware Consolidated Backup -- Virtual Machine restore utility
Version 4.0.0 (build-261974)
Copyright (C) 1998-2010 VMware, Inc.  All rights reserved.
Usage:
vcbResAll -h <url> -u <username> [-p <password>] [-C <datastorecatalog>]
           -s <sourcedir> [-V(0|1)] [-M(0|1)] [-F(0|1)]
           [-m <mode>] [-b <behavior>] [-o <powerop>] [<verbosity>]

              <url> := <hostname>[":"<port>]
 <datastorecatalog> := <filename>
        <sourcedir> := <dirname>
             <mode> := cos|nas
         <behavior> := overwrite|prompt|keep|abort
          <powerop> := on|allon|alloff
        <verbosity> := -L (0-6)

If no password is specified on the command line, you will be prompted for one.

Import Disk Flags:
```

▲ vcbResAll

8. vcbUtil

这是给定 VCB 各种参数的工具，比如 VM 的文件夹或是用户账号口令等参数。

```
root@esxu2:~
[root@esxu2 ~]# vcbUtil
[2010-07-11 16:42:50.019 F4EBEB60 info 'App'] Current working directory: /root
[2010-07-11 16:42:50.061 F4EBEB60 error 'vcbUtil'] Error: Missing command line o
ption.
VMware Consolidated Backup -- Resource Browser and Server Ping
Version 4.0.0 (build-261974)
Copyright (C) 1998-2010 VMware, Inc.  All rights reserved.
Usage:
vcbUtil -h <url> -u <username> [-p <password>] -c <command>
        [<verbosity>]

      <url>    := <hostname>[":"<port>]
 <username>    := <string>
 <password>    := <string>
  <command>    := "ping"|"resourcepools"|"vmfolders"
<verbosity>    := -L (0-6)

If no password is specified on the command line, you will be prompted for one.

Commands:

ping: Attempt to log into the VC server, to verify hostname,
      username and password.
```

▲ vcbUtil

25.2.4 其他常用的工具

除了 ESX 本身的命令之外，有许多 Linux 下有用的文件在 ESX 下也是常用，我们就做一个总汇总。

1．vmk 开头的命令

VMware 上的命令超过百个，我们在这里将做个总汇总。

▶ vmk 开头的命令

- Vmkchdev：遍历 ESX 中的存储设备。
- Vmkerrcode：给出 ESX 上的错误代码。
- Vmkfstools：操作 ESX 中的磁盘。
- Vmkiscsiadm：管理 ESX 下的 iSCSI 设备。
- Vmkiscsid：ESX 下的 iSCSI 的软件连接工具。
- vmkiscsi-tool：配置 ESX 下 iSCSI 的工具程序。
- vmkloader：装载 VMKernel 的程序。
- vmkload_mod：装载 ESX 引导模块的程序。
- vmklogger：管理 ESX Log 的工具。
- vmkmicrocodeintel：ESX 主机中有关 Intel 内核的补丁遍历程序。
- vmkmod-install.sh：安装模块的草稿运行文件。
- vmkmod-preinst.sh：准备安装 ESX 的草稿运行档。
- vmkperf：监视某个事件效能的程序。
- vmkping：使用 VMKernel IP 运行 ping 的程序。
- vmkuptime.pl：理解 ESX 主机引导时间的 Perl 程序。
- vmkvsitools：遍历各项 VMware 设备或设备详细消息的程序。

2．vmware 开头的命令

以下是 vmware-开头的命令。

▶ 以 vmware-开头的命令

- vmware-authd：管理 ESX 的用户账口令及授权。
- vmware-autopoweron：配置 ESX 的自动引导。
- vmware-cmd：运行 VMware 下的 Shell 程序。
- vmware-configcheck：遍历 ESX 下的所有配置文件。
- vmware-hostd：配置及运行 VMware 下的 hostd，即 VMware 的管理功能。
- vmware-hostd-support：获取管理功能的支持。
- vmware-mkinitrd：配置 ESX 下的 initrd 的文件。
- vmware-vim-cmd：使用 vim 中的命令 shell。
- vmware-vimdump：抛弃出 ESX 中 vim 的数据。
- vmware-vimsh：运行 ESX 中的 vimsh。
- vmware-watchdog：监视 hostd。
- vmware-webAccess：配置或激活 ESX 主机的 WebAccess。

结 语

虽然使用 vSphere Client 已经可以完成 90%以上的任务，但 vSphere Client 蓝屏或是需要使用命令大量化的任务时，这些命令将会是你最好的帮手。事实上，要成为 vSphere 的牛人，除了一般的操作之外，最重要的还是脚本文件，而当前最红的云计算，在运用到虚拟机的这一部分，大部分还是利用 vSphere 提供的 API 以及这些脚本文件并发落实的。本章仅给出常用命令，其中许多命令本身就是一部“四书”“五经”，读者们可以在需要时再研读。

读书笔记